



ESTG

2021 SICAP – SISTEMA DE INFORMAÇÃO CADASTRAL E ADMINISTRAÇÃO DE PROPRIEDADE



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

SICAP – SISTEMA DE INFORMAÇÃO CADASTRAL E ADMINISTRAÇÃO DE PROPRIEDADE

Filipe Miguel Dias Rodrigues



INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

Filipe Miguel Dias Rodrigues

SICAP – SISTEMA DE INFORMAÇÃO CADASTRAL E
ADMINISTRAÇÃO DE PROPRIEDADES

Mestrado em Engenharia Informática

Trabalho efetuado sob a orientação de:
Professor Doutor Pedro Miguel Ribeiro Castro
Professor Doutor Joaquim Mamede Alonso

Dezembro de 2021

AGRADECIMENTOS

O projeto que agora termino, corresponde ao culminar de um longo trajeto que não poderia ter percorrido sem a ajuda de várias pessoas a quem muito devo e a quem não poderia deixar de agradecer.

Em primeiro lugar, gostaria de agradecer à minha família, em particular à minha esposa Cátia Gil, que tantas vezes me deu ânimo e que de forma resiliente tantas vezes se ocupou dos nossos três filhos para eu me poder dedicar à vida académica.

Aos meus colegas de mestrado, quero agradecer pela troca de ideias e entreaajuda, mas também pelo companheirismo e momentos de descontração que passamos.

Aos meus orientadores, os sempre presentes Professor Doutor Pedro Castro e Professor Doutor Joaquim Alonso, pela sua inteira disponibilidade e por me ajudarem a desenvolver o pensamento analítico e estruturado que se revelou essencial para o projeto, o meu muito obrigado.

A todos os restantes professores do mestrado em Engenharia Informática do IPVC, o meu agradecimento pelos vossos ensinamentos, eles são a base do meu conhecimento.

O meu último agradecimento vai para os meus pais, Adriano Rodrigues e Celeste Dias. A eles dedico este trabalho. Sem eles não teria chegado onde cheguei, não conseguiria ter feito o que fiz. Não seria quem sou.

RESUMO

O cadastro predial é uma fonte fundamental de informação para o planeamento e gestão do território. O facto de Portugal não possuir um cadastro predial atualizado tem como consequência que não sabemos a quem pertence 20% do território português. Neste contexto, pretende-se com este trabalho contribuir para a disponibilização de ferramentas que possam ser úteis e um auxílio na construção do cadastro predial.

Assim, no âmbito deste projeto, desenvolveu-se a plataforma Web denominada SICAP (Sistema de Informação Cadastral e Administração de Propriedades), a fim de permitir registar dados alfanuméricos e geográficos das propriedades, com informação prevista no SICS (Sistema de Informação Cadastral Simplificado) segundo as diretivas da INSPIRE (Infrastructure for Spatial Information in Europe), que por sua vez suporta a compatibilidade com o standard internacional LADM (Land Administration Domain Model). A plataforma foi ainda complementada com funcionalidades que permitem realizar operações simples de análise e a visualização de métricas e estatísticas através dos dados das propriedades introduzidas.

A especificação da plataforma tem em consideração a abstração da sua finalidade, apostando num conceito mais genérico e modular, garantindo a sua utilidade antes e depois da conclusão do cadastro nacional.

Do ponto de vista de aplicação prática, como exemplo, podemos apontar a utilidade da plataforma no caso de uma entidade privada ou associação, que por ser proprietária de múltiplas parcelas de terreno em diferentes localizações necessitar de uma ferramenta para apoiar a sua gestão. Numa perspetiva de evolução futura, a plataforma SICAP tem as bases para proporcionar a criação de um mercado fundiário online ou banco de terras, que permita a compra, venda, arrendamento ou permuta de parcelas de utilização agrícola e florestal, conseguindo assim a mobilização das terras e a expansão das explorações.

Dezembro de 2021

ABSTRACT

The land registry is a fundamental source of information for the planning and management of the territory. The fact that Portugal does not have an up-to-date land registry means that we do not know who owns 20% of Portuguese territory. In this context, it is intended with this work to contribute to the availability of tools that can be useful and an aid in the construction of the land registry.

Within the scope of this project, the web platform called SICAP (System of Cadastral Information and Property Management) was developed in order to allow the registration of alphanumeric and geographic data of properties, with information provided for in the SICS (Simplified Cadastral Information System) according to INSPIRE directives (Infrastructure for Spatial Information in Europe), which in turn supports the compatibility with the international standard LADM (Land Administration Domain Model). The platform was also complemented with features that allow performing simple analysis operations and the visualization of metrics and statistics through the data of the properties introduced.

The platform specification considers the abstraction of its purpose, focusing on a more generic and modular concept, guaranteeing its usefulness before and after the completion of the national register.

From the point of view of practical application, as an example, we can point out the usefulness of the platform in the case of a private entity or association, which, as the owner of multiple parcels of land in different locations, needs a tool to support its management. In the perspective of future evolution, the SICAP platform has the bases to provide the creation of an online land market or land bank, which allows the purchase, sale, lease, or exchange of parcels for agricultural and forestry use, achieving the mobilization of land and the expansion of farms.

December 2021

LISTA DE ABREVIATURAS / SIGLAS

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
API	Application Programming Interface
AT	Autoridade Tributária
BUPI	Balcão Único do Prédio
CC	Cartão de Cidadão
CGD	Conjunto de dados geográficos
CGPR	Cadastro Geométrico da Propriedade Rústica
CORS	Cross-Origin Resource Sharing
CSS	Cascading Style Sheets
DGT	Direção Geral do Território
FOSS	Free and Open-Source Software
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
ICNF	Instituto da Conservação da Natureza e das Florestas
IFAP	Instituto de Financiamento da Agricultura e Pescas
IGP	Instituto Geográfico Português
IMI	Imposto Municipal sobre Imóveis
INSPIRE	Infrastructure for Spatial Information in Europe
IPCC	Instituto Português da Cartografia e Cadastro
IRN	Institutos dos Registos e Notariado
ISO	International Organization for Standardization
JSON	Javascript Object Notation
KML	Keyhole Markup Language
LADM	Land Administration Domain Model
LTS	Long-term Support (Suporte de Longo Prazo)
NIF	Número de Identificação Fiscal
NIP	Número de Identificação do Prédio
NPM	Node Package Manager
PDM	Plano Director Municipal
REST	Representational State Transfer
RGG	Representação Gráfica Georreferenciada
SICAP	Sistemas de Informação Cadastral e Administração de Propriedades

SICS	Sistema de Informação Cadastral Simplificado
SIG	Sistemas de Informação Geográfica
SiNErGIC	Sistema Nacional de Exploração e Gestão de Informação Cadastral
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SSH	Secure Shell
TI	Tecnologias de Informação
UI	User Interface
UML	Unified Modeling Language
URL	Uniform Resource Locator
WMS	Web Map Service
XML	Extensible Markup Language

ÍNDICE

1. Introdução	1
1.1. Contextualização	1
1.2. Objetivos.....	3
1.3. Método de investigação.....	4
1.4. Estrutura do documento.....	5
2. Sistemas de informação para o cadastro e gestão da propriedade	7
2.1. Os modelos de dados para o desenvolvimento de plataformas e sistemas para o cadastro predial	7
2.1.1. INSPIRE.....	7
2.1.2. LADM	11
3. Programação, fases e engenharia de desenvolvimento da plataforma.....	17
3.1. Objetivos e requisitos da plataforma.....	17
3.2. Modelo de dados da plataforma SICAP.....	18
3.3. A Arquitetura lógica e funcional	22
3.4. As tecnologias	23
3.4.1. Backend	24
3.4.2. Frontend.....	30
3.5. Níveis de acesso à plataforma.....	34
3.6. Funcionalidades da plataforma.....	34
3.6.1. API Restfull.....	34
3.6.2. Autenticação.....	35
3.6.3. Gestão de Utilizadores.....	35
3.6.4. Gestão de Proprietários	37
3.6.5. Gestão de Propriedades	38
3.6.6. Mapa	38

3.6.7. Análise.....	38
3.6.8. Painel de Navegação	39
4. Apresentação e análise de resultados.....	41
4.1. A plataforma.....	41
4.1.1. Iniciar sessão.....	41
4.1.2. Painel de Navegação	41
4.1.3. Configurações	42
4.1.4. Proprietários	44
4.1.5. Propriedades	44
4.1.6. Mapa	53
4.1.7. Análise.....	56
4.2. Testes	57
5. Conclusões e trabalho futuro.....	59
6. Referências	61

ÍNDICE DE FIGURAS

Figura 1. INSPIRE Consolidated UML Model.....	9
Figura 2. Land Administration Domain Model.....	12
Figura 3. Modelo de dados.....	20
Figura 4. Arquitetura do sistema	23
Figura 5. Popularidade das frameworks Angular, React e Vue.....	30
Figura 6. Classificação das frameworks Angular, React e Vue segundo o Git Star Ranking.....	31
Figura 7. Vuex: Fluxo de dados unidirecional.....	32
Figura 8. Casos de uso	36
Figura 9. Tabela users	37
Figura 10. Formulário de Login	41
Figura 11. Painel de Navegação	41
Figura 12. Registo de Utilizadores	42
Figura 13. Layers	43
Figura 14. Proprietários.....	44
Figura 15. Registo de Propriedades: Proprietário	45
Figura 16. Registo de Propriedades: Registo.....	45
Figura 17. Registo de Propriedades: Matriz	46
Figura 18. Registo de Propriedades: Localização.....	47
Figura 19. Registo de Propriedades: Ficheiros	49
Figura 20. Registo de Propriedades: Parcela.....	50
Figura 21. Listagem de Propriedades	51
Figura 22. Botões de consulta.....	51
Figura 23. Consultar Propriedade	52
Figura 24. Consultar Propriedade (continuação).....	52

Figura 25. Consultar Propriedades: Fotografias.....	53
Figura 26. Consultar Propriedades: Documentos.....	53
Figura 27 – Serviços de mapas.....	53
Figura 28. Mapa	54
Figura 29. Legendas Natureza / Valor.....	54
Figura 30. Layers.....	54
Figura 31. PDM de Monção.....	55
Figura 32. CAOP	55
Figura 33. Layers e Filtros.....	55
Figura 34. Análise: Propriedades por proprietário, concelho e natureza	56
Figura 35. Análise: Propriedades por área e por IMI.....	56
Figura 36. Análise: Propriedades por Valor Patrimonial.....	57

ÍNDICE DE TABELAS

Tabela 1. Requisitos Funcionais	17
Tabela 2. Comparativo entre as frameworks Angular, React e Vue.....	31
Tabela 3. Níveis de acesso dos utilizadores	36

1. INTRODUÇÃO

1.1. Contextualização

“...um país que nem sequer se conhece dificilmente pode desenvolver-se, ou desenvolver-se de forma ordenada e harmoniosa.” (de Beires, Amaral, & Ribeiro, 2013). O facto de Portugal não possuir um cadastro predial atualizado tem como consequência que não saibamos a quem pertence 20% do nosso país (de Beires, Amaral, & Ribeiro, 2013).

Face a este número, a realização de um cadastro predial torna-se uma necessidade inadiável. É necessário responder de forma rápida a questões como: Quem são os donos da propriedade? Onde está localizada? Qual a sua dimensão? Como chegar a esses proprietários?

No passado foram desencadeadas duas iniciativas de reconhecimento do território: o Cadastro Geométrico da Propriedade Rústica (CGPR) e o Cadastro Predial e Sistema Nacional de Exploração e Gestão de Informação Cadastral (SiNErGIC).

O CGPR foi uma Iniciativa que ocorreu há algumas décadas atrás, avançando de sul para norte dando prioridade à área do macrofúndio, contudo este processo não obteve a continuidade necessária à sua total implementação ao iniciar o levantamento nas áreas de minifúndio em zonas de relevo acentuado, provavelmente por se ter revelado uma tarefa muito mais complexa de realizar, tendo possivelmente decaído a execução por falta de incentivo, tempo e verbas para a continuidade do mesmo.

Mais recentemente foi implementado o SiNErGIC para o qual, apesar de já contar com alguns anos e investimentos significativos, verifica-se que os resultados obtidos não conseguiram atingir uma cobertura significativa do território e não se observam dinâmicas para atingir os objetivos propostos.

Ambas as iniciativas partiram do conceito tradicional de operação sistémica feita no terreno diretamente pelo Estado.

Atualmente encontra-se em curso o Cadastro Simplificado, através do Sistema de Informação Cadastral Simplificado e do Balcão Único do Prédio (BUPi), com uma abordagem diferente.

Este cadastro assenta num procedimento de representação gráfica georreferenciada (RGG), que se diferencia por ser feito pelos titulares tendo por base ortofotomapas disponíveis através de uma plataforma eletrónica, com recurso a um técnico habilitado municipal.

O Cadastro Simplificado é marcado pela interoperabilidade, que se vem opor à criação de uma base de dados integrada de informação pública, e que prima por não solicitar ao cidadão aquilo que a administração já conhece.

Este conhecimento é fruto da recolha de informação: da DGT (ortofotomapas); do IFAP (Parcelário); do ICNF (Florestas e Matas Nacionais); da AT (matrizes cadastrais); do IRN (registo predial); do Centro Geoespacial do Exército (cartografia); das Estradas de Portugal (caminhos); das Associações Florestais (levantamentos efetuados) e dos Municípios (informação diversa incluindo baldios).

O Sistema de Informação Cadastral Simplificado mantém o foco no desenvolvimento em código aberto, mantendo a lógica de interoperabilidade entre todas as fontes de informação identificadas ou a identificar, desenvolvendo de forma continuada informação em formato aberto para uso público (Open Data) (G. do Sec. de Estado dos Assuntos Parlamentares, 2018).

No último relatório do governo acerca do Cadastro Simplificado, é referido o sucesso com que o mesmo está a ser implementado nos 10 concelhos pilotos, onde o Governo atingiu os objetivos a que se propôs e excedeu o inicialmente previsto. É também sugerido que o alargamento deve ocorrer em simultâneo em todos os municípios onde não exista outra forma de cadastro (G. do Sec. de Estado dos Assuntos Parlamentares, 2018).

O Cadastro Simplificado parece ser a solução para Portugal finalmente poder contar com um cadastro predial atualizado, que irá definir com rigor as características de cada prédio rústico ou urbano, designadamente a quem pertence, a sua localização, configuração, limites e áreas do imóvel, atribuindo-lhe um Número de Identificação de Prédio (NIP) que passará a ser o número de identidade unívoco do prédio. No entanto, esta realidade ainda tardará uns anos a estar completamente implementada em todo o território nacional.

É neste quadro geral que identificamos a vantagem da criação de uma plataforma WebSIG, que possa dar um contributo para apoiar ou facilitar a execução do

cadastro nacional através de uma ferramenta que permita a gestão de propriedades de uma entidade privada/particular assente nos conceitos e regras definidas para o cadastro simplificado, permitindo assim a interoperabilidade com o sistema nacional, e incluindo um conjunto de funcionalidades específicas, adequadas e necessárias para uma administração privada ou particular e estruturada de forma a contemplar as diversas diretivas, standards e modelos definidos na área da gestão de propriedades.

A plataforma desenvolvida no contexto deste trabalho, permite registar dados alfanuméricos e geográficos das propriedades com informação prevista no SICS (Sistema de Informação Cadastral Simplificado), segundo as diretivas da INSPIRE (Infrastructure for Spatial Information in Europe), que por sua vez suporta a compatibilidade com o standard internacional LADM (Land Administration Domain Model) (I. T. W. Group, 2014). Além da gestão da componente alfanumérica e geográfica, a plataforma foi complementada com funcionalidades que oferecem a possibilidade de realizar operações de análise (espacial) e visualização de métricas e estatísticas através da análise dos dados das propriedades introduzidas.

A especificação da plataforma, desenvolvida com tecnologias e software de código aberto e de utilização gratuita (Free and Open Source Software - FOSS) tem em consideração a abstração da sua finalidade, apostando num conceito e abordagem genérica e modular, garantindo a sua utilidade antes e depois da conclusão do cadastro nacional.

1.2. Objetivos

O objetivo geral da plataforma centra-se no desenvolvimento de uma plataforma com utilidade para a gestão de propriedades por uma entidade privada ou particular, facilitando a sua administração garantindo a compatibilidade com os sistemas/standards existentes, sendo um exemplo de aplicação prática o caso de uma entidade privada ou associação, que por ser proprietária de múltiplas parcelas de terreno em diferentes localizações, necessita uma ferramenta para fazer a sua gestão de forma simples assegurando que esse registo é integrável/exportável para o sistema nacional oficial de registo do cadastro.

Como um dos focos primordiais é manter uma compatibilidade entre os dados da plataforma e os do cadastro oficial, será possível fazer os registos (RGG) das

propriedades de modo imediato na plataforma e mais tarde exportá-los para o cadastro quando estiver em execução no município em questão. Também contemplará operações para importar os dados para a plataforma, caso o cadastro daquela zona já tenha sido efetuado.

O desenvolvimento desta plataforma assume-se ainda como uma base para a criação de outras ferramentas de interesse na área da administração de parcelas e promoção de instrumentos fundamentais ao desenvolvimento económico, como é o exemplo prático da criação de um mercado fundiário online ou banco de terras, que permita a compra, venda, arrendamento ou permuta de parcelas de utilização agrícola e florestal, conseguindo assim a mobilização das terras e a expansão das explorações.

1.3. Método de investigação

A investigação foi planeada, desenvolvida e validada utilizando um método de investigação apropriado, recolhendo dados para análise, descobrindo nova informação e criando um melhor entendimento do tópico.

Tratando-se este trabalho de um projeto de desenvolvimento de ferramentas na área dos sistemas de informação com foco na transformação digital, o método de investigação utilizado, por se julgar o mais apropriado, foi o *Design Science Research*.

Este método foca-se no desenvolvimento de artefactos inovadores que estendem os limites atuais das Tecnologias de Informação (TI) para impulsionar o desempenho das empresas e organizações (Cruz & da Cruz, 2020).

Após a identificação do problema e da motivação, pretendeu-se a criação de um artefacto rigorosamente definido, útil e inovador para resolver um problema específico no domínio apresentado. Para isso, teve que existir um bom entendimento do estado da arte na área do problema e a definição de objetivos, neste caso qualitativos, que permitiram chegar à solução do problema através de uma interação entre pessoas, organizações e tecnologias de informação.

O resultado poderia ser a criação de uma solução única e inovadora para resolver um problema não resolvido, ou então apresentar uma solução mais eficiente e eficaz para um problema já anteriormente resolvido.

Na produção do artefacto, as atividades do ciclo Construir->Avaliar do método *Design Science Research* geraram um processo interativo que ocorreu diversas vezes. Através dessas interações, baseadas em métodos rigorosos de construção e avaliação, o artefacto demonstrou a sua utilidade, qualidade e eficácia levando a investigação ao encontro dos objetivos estabelecidos.

Para chegar a esta conclusão, a avaliação foi essencial, verificando quão bem o artefacto suportou a solução do problema, avaliando-o em termos de funcionalidade, integridade, consistência, precisão, desempenho, confiabilidade, usabilidade, adequação aos requisitos e outras qualidades relevantes.

Por fim, na forma de comunicação para dar a conhecer os resultados da investigação, teve-se em conta que a sua apresentação deve ser feita a pensar numa audiência quer orientada para a tecnologia quer orientada para a gestão (Cruz & da Cruz, 2020).

1.4. Estrutura do documento

Após o capítulo da introdução, este trabalho de mestrado desenvolve-se em mais quatro capítulos.

No segundo capítulo são apresentados modelos de dados europeus e internacionais para o desenvolvimento de plataformas e sistemas de informação para o cadastro predial e gestão de propriedades. Os modelos apresentados resultam de um estudo do estado da arte sobre o tema.

No terceiro capítulo são descritos os objetivos e requisitos, o modelo de dados, a arquitetura, as tecnologias utilizadas, os níveis de acesso e as funcionalidades da plataforma.

No quarto capítulo é feita uma apresentação da plataforma e análise dos resultados.

Por último, no quinto capítulo é descrita a conclusão do trabalho e apresentadas algumas propostas para trabalhos futuros.

2. SISTEMAS DE INFORMAÇÃO PARA O CADASTRO E GESTÃO DA PROPRIEDADE

2.1. Os modelos de dados para o desenvolvimento de plataformas e sistemas para o cadastro predial

Conforme referido anteriormente, um dos objetivos no desenvolvimento desta plataforma é a interoperabilidade dos conjuntos de dados geográficos (CDG) com as plataformas já existentes de cadastro nacional e europeu, pelo que se torna pertinente a utilização de um modelo de dados que siga as diretivas INSPIRE e LADM.

A diretiva INSPIRE foi elaborada com o intuito de criar a infraestrutura europeia de informação geográfica, incidindo em informação da responsabilidade das instituições públicas dos estados-membros, cujo objetivo final consiste em facultar o acesso aos serviços de dados geográficos através do geoportal INSPIRE. A adesão a esta diretiva, implica que seja respeitado um conjunto de normas que permitem a interoperabilidade de dados e serviços de dados geográficos através da Europa (Direção-Geral do Território, 2021).

A Infraestrutura Europeia de Informação Geográfica tem como finalidade a disponibilização aos utilizadores, de serviços integrados de informação espacial presentes numa rede distribuída de base de dados, com uma base em standards e protocolos comuns assegurando a sua compatibilidade (Direção-Geral do Território, 2021).

2.1.1. INSPIRE

A diretiva INSPIRE inclui a definição das condições necessárias para a criação da Infraestrutura Europeia de Informação Geográfica, com o objetivo de facultar os meios dos cidadãos europeus poderem descobrir, utilizando meios disponibilizados na Internet, informação (geográfica) útil numa diversidade de áreas temáticas, permitindo ainda uma mais fácil partilha de informação entre entidades públicas (e o público em geral).

Nesse sentido, os princípios da iniciativa INSPIRE refletem as pretensões a conseguir com a implementação da diretiva (Direção-Geral do Território, 2021):

- ▮ Os dados devem ser recolhidos uma vez e atualizados no nível em que tal possa ser realizado com maior eficácia;
- ▮ A informação geográfica proveniente de diferentes fontes, deve poder ser combinada de forma transparente, através da Europa, e partilhada por diversos utilizadores e aplicações;
- ▮ Deve ser possível a partilha de informação recolhida a um determinado nível com todos os outros níveis, detalhada para análises pormenorizadas e geral para objetivos estratégicos;
- ▮ A informação geográfica de suporte à atividade governamental, a todos os níveis, deve ser abundante e disponível sob condições que não restrinjam o seu uso generalizado;
- ▮ A informação geográfica disponível, tem que ser facilmente identificável, devendo ser fácil analisar a sua adequabilidade para um determinado uso bem como as respetivas condições de acesso e utilização;
- ▮ A informação geográfica deve tornar-se cada vez mais perceptível e fácil de interpretar por se encontrar devidamente documentada e por poder ser visualizada no contexto adequado, selecionado de forma amigável para o utilizador.

No que se refere ao foco deste projeto, é de mencionar que a diretiva INSPIRE não visa a harmonização de conceitos de propriedade ou direitos relativos às parcelas. O seu foco são apenas os aspetos geométricos apresentados nos sistemas dos Estados-Membros (I. T. W. Group, 2014).

O modelo de dados da INSPIRE, representado na Figura 1 (European Commission, 2021) é composto, no que respeita ao registo do cadastro, pela classe central *CadastralParcel* e três classes auxiliares: *CadastralZoning*, *CadastralBoundary* e *BasicPropertyUnit*.

Neste modelo de dados, existem alguns atributos gerais comuns a todos os objetos espaciais. Cada uma das classes contém um identificador *INSPIRE* (*inspireID*) e um conjunto de atributos temporais:

- ▮ *beginLifespanVersion* e *endLifespanVersion* estão relacionados com o tempo de vida do objeto espacial no conjunto de dados espaciais (ponto de vista geográfico / conjunto de dados);

|| *validFrom* e *validTo* estão relacionados com o tempo de vida da entidade no mundo real (ponto de vista jurídico).

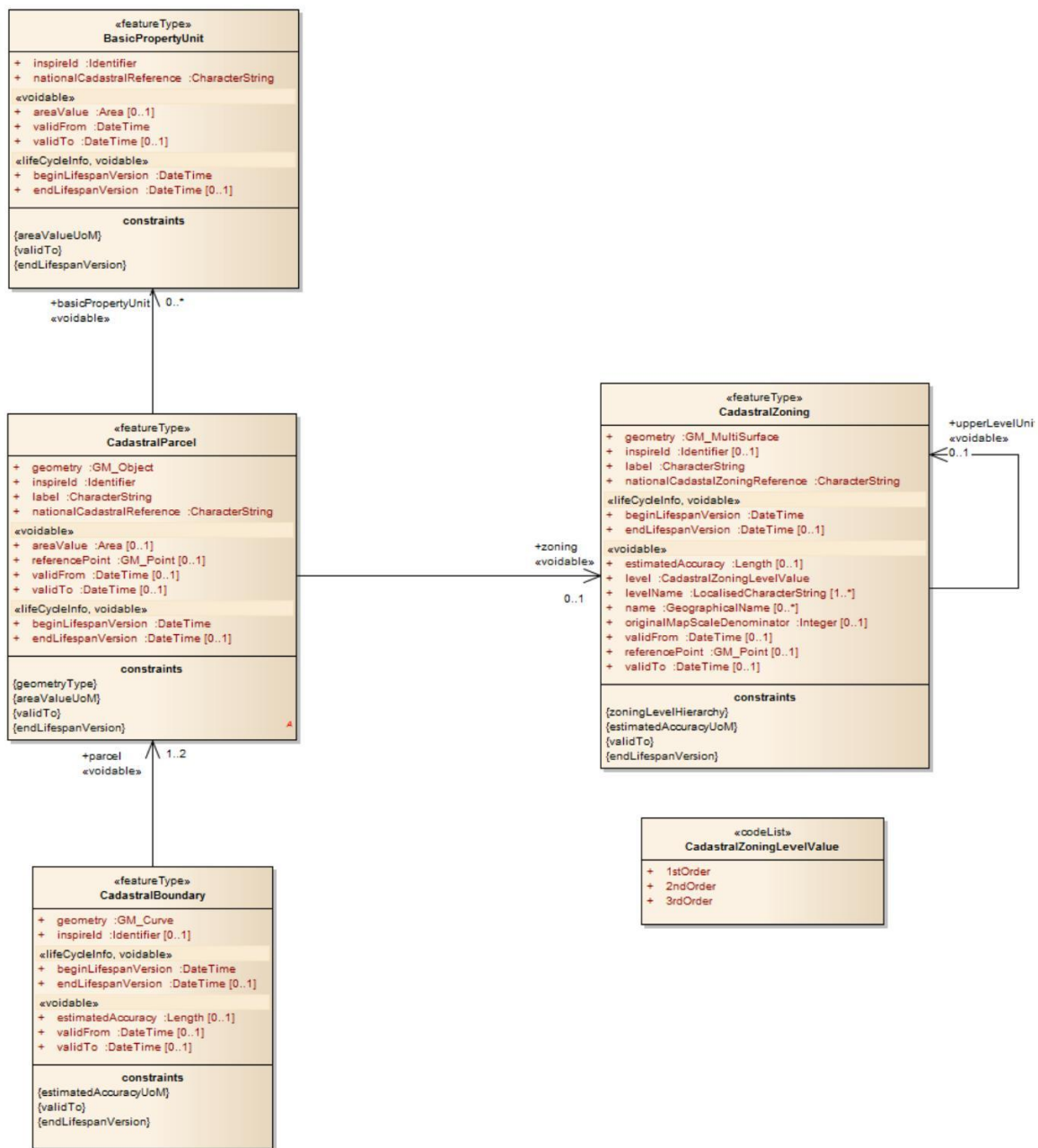


Figura 1. INSPIRE Consolidated UML Model¹

¹ Figura disponível em:

<https://inspire.ec.europa.eu/data-model/approved/r4618-ir/html/index.htm?goto=2:1:3:1:7204>

Na classe *CadastralParcel* as parcelas cadastrais são definidas pela diretiva INSPIRE como "áreas definidas por registos cadastrais ou equivalente" e quanto possível devem constituir uma partição do território nacional. A parcela cadastral deve ser uma área única de superfície terrestre (terra e/ou água), sob direitos de propriedade definidos pela legislação nacional.

As parcelas cadastrais (*CadastralParcel*) contemplam os seguintes atributos:

- ▮ Uma geometria
- ▮ Uma referência cadastral nacional
- ▮ Um valor de área
- ▮ Um ponto de referência
- ▮ Um rótulo (etiqueta)

As zonas cadastrais (*CadastralZoning*) são as áreas intermediárias (como concelhos, freguesias, secções, quarteirões, etc.) utilizadas para dividir o território nacional em parcelas cadastrais.

As zonas cadastrais incluem os seguintes atributos:

- ▮ Uma geometria
- ▮ Uma referência de zona cadastral nacional
- ▮ Um nome, caso exista
- ▮ Um nível na hierarquia cadastral nacional e nome desse nível
- ▮ Um ponto de referência
- ▮ Um rótulo (etiqueta)
- ▮ Atributos de meta-dados: denominador da escala do mapa original e precisão estimada

Nos casos em que são fornecidas zonas cadastrais, as parcelas cadastrais devem pertencer a uma zona do nível mais baixo. Quando existem vários níveis de zoneamento, deve assegurar-se que as unidades de nível superior sejam compostas pelas de nível inferior.

No contexto INSPIRE, os limites cadastrais (*CadastralBoundary*) devem ser disponibilizados pelos Estados-Membros onde a informação de precisão posicional absoluta é registada para o limite cadastral (atributo da precisão estimada).

Os limites cadastrais têm os seguintes atributos adicionais:

- ▮ Uma geometria
- ▮ Atributos de meta-dados: precisão estimada

Por último, as unidades básicas de propriedade (*BasicPropertyUnit*) são as unidades básicas que são registadas nas conservatórias do registo predial ou equivalentes. São definidas pelos direitos de posse e direitos de propriedade homogêneos e podem consistir em um ou mais lotes adjacentes ou separados.

No contexto INSPIRE as unidades básicas de propriedade devem ser utilizadas por países onde as referências cadastrais são atribuídas apenas às unidades básicas de propriedade e não às parcelas.

As unidades básicas de propriedade têm os seguintes atributos adicionais:

- ▮ Uma referência cadastral nacional
- ▮ Um valor de área

Relativamente ao modelo de dados para as parcelas cadastrais definido na diretiva INSPIRE, é ainda de referir que foi preparado de forma a suportar a compatibilidade com o padrão internacional *Land Administration Domain Model* (LADM) que fornece um contexto mais amplo para as parcelas cadastrais INSPIRE porque o LADM inclui informações adicionais sobre direitos (vinculados à legislação nacional) e proprietários, que estão fora do propósito direto da INSPIRE.

2.1.2. LADM

A *Land Administration Domain Model* (LADM) representado na Figura 2 é um standard internacional definido na ISO19152 (ISO, 2012) e é baseado nas relações entre pessoas e propriedades (Lemmen, van Oosterom, & Bennett, 2015).

O LADM estimula o desenvolvimento de aplicações de software e sistemas de administração de propriedades que suportam objetivos sustentáveis, facilitando o intercâmbio de dados entre sistemas distribuídos, permitindo por exemplo combinar dados recebidos de entidades como o Registo Predial, as Finanças e a Direção Geral do Território.

O LADM é um modelo conceptual e não uma especificação de produto (Lemmen, van Oosterom, & Bennett, 2015). O Standard é um modelo abstrato com três

módulos correspondentes: (i) às partes (pessoas e organizações), (ii) aos direitos, deveres e restrições (direitos de propriedade) e (iii) às unidades espaciais (parcelas e frações).

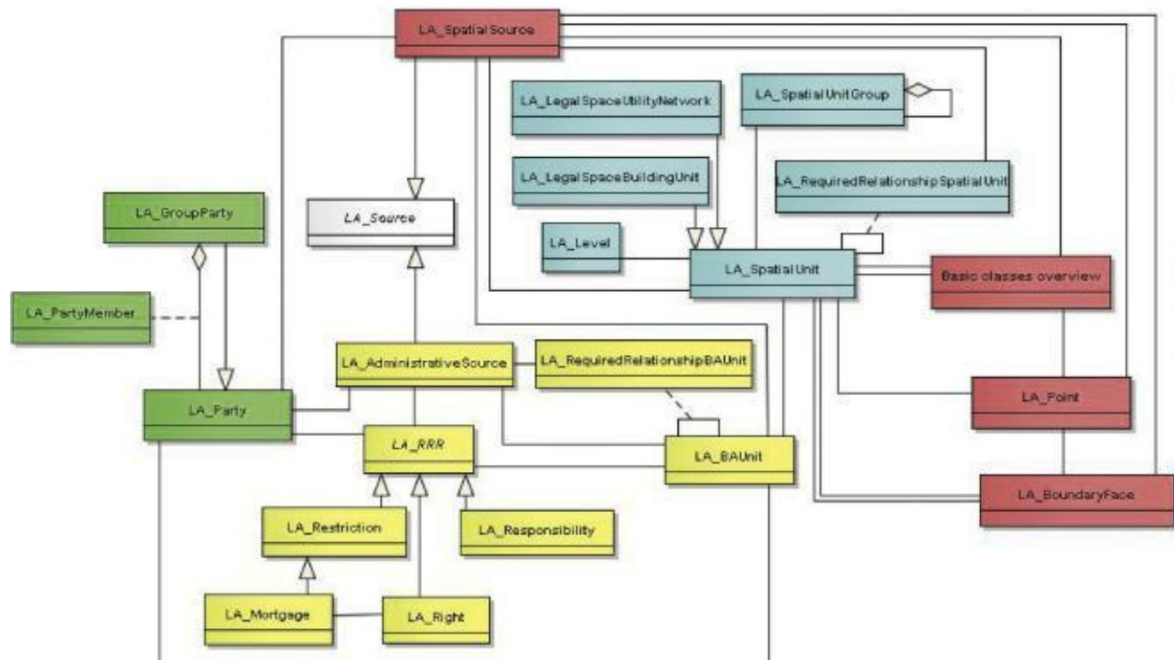


Figura 2. Land Administration Domain Model²

No desenvolvimento do modelo foram consideradas as características que um standard para a administração de propriedades deve ter, como ser útil na prática, ser adaptável e aplicável às situações locais, ajustando-se facilmente às especificidades de cada realidade dos estados onde venha a ser aplicado, e a tecnologia adotada ser suficientemente flexível para que o sistema possa adaptar-se e crescer para permitir a evolução do modelo de acordo com os requisitos e necessidades que venham a ser identificadas no futuro.

Do ponto de vista global, o modelo foi criado para responder aos requisitos na administração de propriedades, tendo em conta que essas necessidades variam de local para local. Com efeito, existem diferenças nos procedimentos de cadastro, em que alguns países baseiam os seus registos numa base fiscal e outros em títulos

² Figura disponível em: <https://www.sciencedirect.com/science/article/pii/S0264837715000174>

de registo, sendo, por outro lado, alguns sistemas centralizados e outros descentralizados. Considerando esta variabilidade de situações, o modelo apresentado é suficientemente flexível para se adaptar às diferentes realidades.

É ainda de referir que o sistema pode também ser desenhado para suportar níveis elevados de transparência, como permitir registar o nome e a função das pessoas que intervêm na manutenção dos dados.

O modelo LADM consiste em três módulos principais (Figura 2) constituídos por um conjunto de elementos (classes), que são: *Party package* (elementos com fundo verde), o *Administrative package* (elementos com fundo amarelo), e o *Spatial Unit package* (elementos com fundo azul) com o sub-módulo *Representation e Survey* (elementos com fundo vermelho).

O primeiro módulo, Party Package, agrega a informação sobre as partes, que pode ser um sujeito ou organização que desempenha um papel com direitos sobre a propriedade. Neste módulo, a classe principal é a classe LA_Party que diz respeito a uma parte, com a sua especialização La_GroupParty que pode ser qualquer conjunto de partes formando no seu conjunto uma entidade.

O *Administrative package*, inclui informação acerca dos direitos, restrições e responsabilidades sobre uma propriedade. Contém a classe abstrata LA_RRR com três subclasses concretas:

- LA_Right, que permite registar os direitos sobre as propriedades;
- LA_Restriction para registar as restrições, como uma hipoteca por exemplo;
- LA_Responsability que diz respeito às responsabilidades, que podem ser obrigações formais ou informais;

contém também a classe La_BAUnit (*Basic Administrative Unit*) que se trata de uma unidade administrativa que pode agregar uma ou mais unidades espaciais. Um exemplo de uma unidade básica administrativa com duas unidades espaciais é um apartamento que contém uma garagem.

Por último, o Spatial Unit package, que diz respeito aos dados geográficos da propriedade, é composto pelas classes:

- || LA_SpatialUnit, que pode ser representada como texto (“desde o caminho até ao rio”), um ponto ou multiponto, uma linha ou multilinha, representando uma área;
- || LA_SpatialUnitGroup, que é um conjunto de SpatialUnits
- || LA_Level, que é uma coleção de SpatialUnits com coerência temática ou topológica
- || LA_LegalSpaceBuildingUnit que contém informação sobre a área legal construtiva

e requer a LA_RelationshipSpatialUnit que faz a ligação com o sub-módulo *Representation e Survey*, que agrega informação como pontos, distâncias e limites.

Como dito anteriormente, o LADM é um modelo conceptual abstrato, ajustável às situações locais. Em Portugal, os dados que caracterizam os prédios, definidos pelo Instituto Português da Cartografia e Cadastro (IPCC), que, entretanto, foi substituído pelo Instituto Geográfico Português (IGP) e que se mantém até à data são (Silva, 2012):

- || NIP - Número de Identificação Predial
- || Distrito
- || Concelho
- || Freguesia
- || Localização geográfica (coordenadas geográficas)
- || Configuração geográfica, que é representada por uma linha fechada que passa pelos pontos que representam os limites de propriedade
- || Área do prédio

Este conjunto de dados são perfeitamente compatíveis com o módulo *Spatial Unit Package do LADM*.

Em síntese, da revisão bibliográfica realizada sobre os requisitos do modelo de dados para a implementação de um sistema de informação cadastral e de administração de propriedades, sobressai a necessidade de o mesmo contemplar uma estrutura que possibilita o registo de dados necessários à gestão do dia a dia das propriedades, mas também incorporar as regras e princípios dos sistemas

oficiais e dos modelos standard definidos na europa e internacionalmente para a gestão cadastral.

Considerando estas premissas, o desenvolvimento da plataforma foi conceptualizado de forma a integrar os conceitos e regras definidas para o Cadastro Simplificado (BUPi), a diretiva INSPIRE e o LADM.

3. PROGRAMAÇÃO, FASES E ENGENHARIA DE DESENVOLVIMENTO DA PLATAFORMA

O desenvolvimento da plataforma SICAP centrou-se na implementação de uma plataforma Web que facultasse as funcionalidades para a criação de um repositório de dados cadastrais e para facultar um meio para a gestão, consulta, comunicação e exportação da informação recolhida.

Nos pontos seguintes apresenta-se a plataforma Web desenvolvida no que se refere (aos seus objetivos e requisitos funcionais), modelo de dados, arquitetura e tecnologias utilizadas, níveis de acesso, assim como as funcionalidades implementadas.

3.1. Objetivos e requisitos da plataforma

Do ponto de vista global, o objetivo da plataforma centrou-se na disponibilização de um portal Web que permita o armazenamento e facilite a visualização, a consulta, a pesquisa e a análise dos dados referentes ao cadastro da propriedade.

De acordo com o objetivo geral definido para a plataforma com funcionamento em ambiente Web, consideraram-se os seguintes requisitos para a estrutura de dados e funcionalidades a incluir no desenvolvimento da aplicação:

- ┃ Um modelo de dados que proporcione o registo da informação essencial para a gestão das propriedades e compatível (ou interoperável) com os modelos existentes e standards atuais;
- ┃ Funcionalidades que facultem:

Tabela 1. Requisitos Funcionais

RF01	gerir (inserir, consultar, atualizar ou remover) os dados referentes a cada utilizador
RF02	Estabelecer o controlo do acesso à plataforma de acordo com os perfis definidos
RF03	gerir (inserir, consultar, atualizar ou remover) os dados referentes a cada proprietário
RF04	gerir (inserir, consultar, atualizar ou remover) os dados referentes a cada propriedade
RF05	visualização e manipulação dos limites dos prédios num mapa interativo

RF06	gerir (inserir, consultar, atualizar ou remover) fotografias referentes às propriedades
RF07	gerir (inserir, consultar, atualizar ou remover) documentos referentes às propriedades
RF08	gerir (inserir, consultar, atualizar ou remover) novas camadas (layers)
RF09	acrescentar novas camadas e filtros a um mapa interativo
RF10	operações de pesquisa
RF11	operações de análise

Nota: As funcionalidades são dependentes do perfil de acesso do utilizador

3.2. Modelo de dados da plataforma SICAP

No contexto do desenvolvimento de um sistema de informação, e em particular no que se refere à plataforma SICAP desenvolvida neste projeto, a definição do modelo de dados é um dos passos fundamentais para a adequação da plataforma aos requisitos pretendidos. Conforme indicado anteriormente, a plataforma pretende dar resposta às necessidades de gestão/administração de propriedades de um utilizador final, garantindo ainda a compatibilidade com os modelos de dados dos sistemas atuais e de standards europeus e internacionais.

A elaboração do modelo de dados da plataforma, assentou numa abordagem que permite definir e classificar os objetos geográficos segundo uma determinada estrutura.

O modelo de dados proposto na Figura 3, que é a base da especificação da plataforma, está correlacionado com o modelo da diretiva INSPIRE no que toca à informação geográfica da parcela, acrescentando os campos definidos pelo IGP para caracterizar os prédios.

No entanto, como descrito anteriormente, a diretiva INSPIRE foca-se apenas na informação geográfica e não visa a harmonização de conceitos de propriedade ou direitos relativos às parcelas (I. T. W. Group, 2014). Foram então utilizados os conceitos do LADM para criar as classes necessárias para facultar o conjunto mínimo de informação que se considerou essencial para uma boa gestão de propriedades.

O modelo resultante da análise compreende um conjunto de cinco grupos principais de dados referentes: (i) ao prédio, que corresponde a uma propriedade com uma identificação única; (ii) registo, que diz respeito aos dados normalmente associados à certidão de registo da conservatória; (iii) proprietário, podendo ser um sujeito ou organização; (iv) matriz, que diz respeito aos dados associados à caderneta predial das finanças; (v) informação complementar sobre prédios como fotografias e documentos.

Assim, temos uma classe central referente ao prédio com os atributos:

- ▮ id - Código único gerado pela plataforma para identificação interna do prédio
- ▮ idNacional - Número de Identificação do Prédio (NIP), que é o número de identidade unívoco do prédio. Este número é atribuído no cadastro
- ▮ descricao - Descrição textual do prédio, que poderá ser por exemplo: Terreno para construção ou Moradia unifamiliar
- ▮ geometria - Geometria com a configuração do prédio, equivalente à sua área de implantação
- ▮ pontoReferencia - Ponto geométrico com as coordenadas correspondentes à localização do prédio
- ▮ area - Área do prédio expressa em metros quadrados
- ▮ imagemCapa – Uma fotografia do prédio (utilizada na identificação geral do prédio)
- ▮ areaImplatacao – Área de implantação. No caso do prédio se encontrar edificado, representa a área ocupada pela base da construção
- ▮ areaConstrução – Área de construção. No caso do prédio se encontrar edificado, representa a área ocupada pela totalidade da construção. Por exemplo, no caso de uma moradia de dois andares seria a soma da área dos dois pisos.
- ▮ dataAquisicao – Data de aquisição do prédio
- ▮ valorAquisicao – Valor de aquisição do prédio
- ▮ dataAvaliacao – Data da avaliação
- ▮ avaliacao – Valor da avaliação do prédio
- ▮ validoDesde - Data de inserção/criação do registo

- || validoAte - Data até quando o registo do prédio é válido, o que pode deixar de acontecer quando por exemplo for anexo a outro e dê origem a um novo prédio com um novo NIP

Da mesma forma que o Sistema de Informação Cadastral Simplificado prima por não solicitar ao cidadão aquilo que a administração já conhece, também a plataforma utiliza o máximo de informação já disponível na caderneta predial da matriz das finanças e da certidão de registo da Conservatória Notarial.

Temos então uma classe Registo cujos atributos são os presentes numa certidão permanente:

- || id - Código único gerado pela plataforma para identificação interna do registo
- || numeroRegisto - Número da certidão permanente do prédio
- || conservatoria - Nome da Conservatória onde foi efetuado o registo
- || direitos - Direitos sobre a propriedade
- || restricoes - Restrições sobre a propriedade, como uma hipoteca por exemplo
- || responsabilidades - Responsabilidades sobre a propriedade

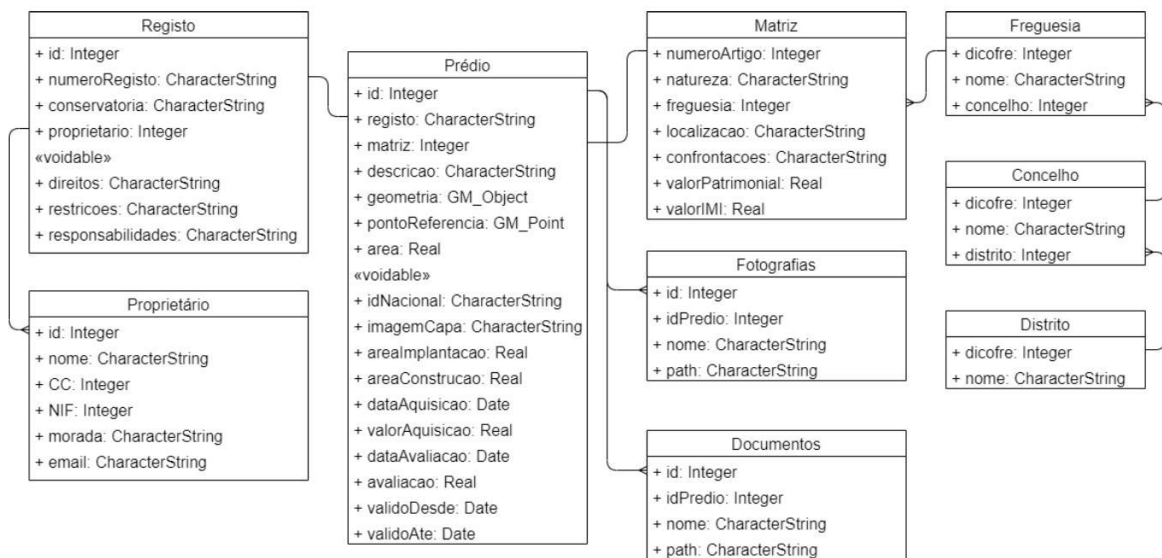


Figura 3. Modelo de dados

A classe Registo é complementada pela classe Proprietário com os dados do proprietário, que pode ser apenas um, ou vários:

- || id - Código único gerado pela plataforma para identificação interna do proprietário
- || nome - Nome do proprietário

- ┆ CC - Número de Cartão de Cidadão do proprietário
- ┆ NIF - Número de Identificação Fiscal do proprietário
- ┆ morada - Morada do proprietário
- ┆ email - Email do proprietário

A classe Matriz contém os atributos que estão presentes na caderneta predial das finanças, com mais um atributo relativo ao valor anual a pagar de IMI, que está presente na caderneta predial de forma implícita através do valor patrimonial, mas não de forma explícita:

- ┆ numeroArtigo - Número do artigo (prédio) inscrito na matriz das finanças
- ┆ natureza - A natureza de um prédio pode ser artigo Rústico ou Urbano
- ┆ localizacao - Localização do prédio descrita textualmente
- ┆ confrontacoes - Nome dos confrontantes a Norte, Sul, Este e Oeste
- ┆ valorPatrimonial - Valor patrimonial do prédio
- ┆ valorIMI - Valor anual a pagar de IMI

As classes Freguesia, Concelho e Distrito permitem a hierarquização da divisão administrativa utilizada na classe Matriz e integra os seguintes dados:

- ┆ dicofre - Identificador único da Freguesia, Concelho e Distrito
- ┆ nome - Nome da Freguesia, Concelho ou Distrito

As classes Fotografias e Documentos, que facultam a associação de informação adicional sobre os prédios sobre a forma de ficheiros, contém os atributos:

- ┆ id – Código único gerado pela plataforma para identificação interna do ficheiro
- ┆ idPredio – Identificador do prédio a que corresponde o ficheiro
- ┆ nome – Nome do ficheiro
- ┆ path – Caminho para a localização do ficheiro no servidor

Este modelo permite uma harmonização com o modelo INSPIRE facilmente alcançável com apenas uma ressalva que se prende com o nationalCadastralReference, atributo que neste modelo de dados é equivalente ao idNacional ou NIP.

Com efeito a diretiva INSPIRE aponta o atributo como sendo de preenchimento obrigatório, mas em Portugal, por não existir ainda um cadastro completo, existem

ainda muitas parcelas sem NIP atribuído. Contudo, esta situação por estar prevista no modelo de dados da plataforma, não irá inviabilizar a sua utilização futura. Neste momento, em Portugal as propriedades são identificadas pelo seu número de registo da conservatória ou pelo número de artigo da matriz, mas qualquer um desses números não é equivalente ao que aqui se pretende.

Existe também a possibilidade de criar uma correlação entre as classes referentes às Freguesias, Concelhos e Distritos com a classe CadastralZoning da diretiva INSPIRE, correspondendo a zonas de primeira, segunda e terceira ordem respetivamente, criando uma maior harmonização entre os dois modelos.

A informação sobre os utilizadores da plataforma é guardada numa tabela da base de dados de nome “users”, que é a única tabela que não está presente no modelo de dados da Figura 3 por não dizer respeito diretamente à administração das propriedades, mas sim ao controlo de acesso do software.

A vantagem de ter toda esta informação coletada numa só ferramenta facilita a gestão porque dá resposta às necessidades do utilizador, sejam elas de carácter geográfico, legal ou contributivo.

3.3. A Arquitetura lógica e funcional

Considerando os objetivos da plataforma SICAP, a definição e especificação da arquitetura lógica e tecnológica teve em consideração as seguintes características e requisitos no desenvolvimento: i) incluir na plataforma as estruturas de dados necessárias para armazenar, organizar e aceder aos dados, quer alfanuméricos quer geográficos, num sistema único e integrado, assim como para gerir os perfis de utilizador e privilégios de acesso; ii) proporcionar uma interface gráfica simples e fácil de utilizar; iii) acessível a partir de um navegador Web sem necessitar de instalação de software adicional; e (iv) recorrer a software gratuito e de código aberto (Free Open Source Software) no sentido de facultar uma solução com custos reduzidos de desenvolvimento e de manutenção. Salienta-se ainda que na seleção do software para a implementação do sistema, foram tidos em conta fatores como a abrangência das funcionalidades a implementar, o conhecimento e experiência adquirida em projetos anteriores, a compatibilidade com standards, o tipo de licenciamento, a existência de recursos de suporte, a dimensão da comunidade de utilizadores e a disponibilidade de documentação e exemplos de implementação.

No sentido de criar um sistema modular para facilitar a manutenção e o desenvolvimento futuro de novas funcionalidades, a plataforma assenta numa arquitetura cliente servidor, com uma camada de Frontend (voltada para o utilizador) em execução como um cliente avançado que se comunica com um Backend (servidor) que fornece a lógica de negócio, fluxo de trabalho, integração e serviço de dados.

No esquema da Figura 4 podemos observar as camadas que compõe a plataforma e as suas ligações:

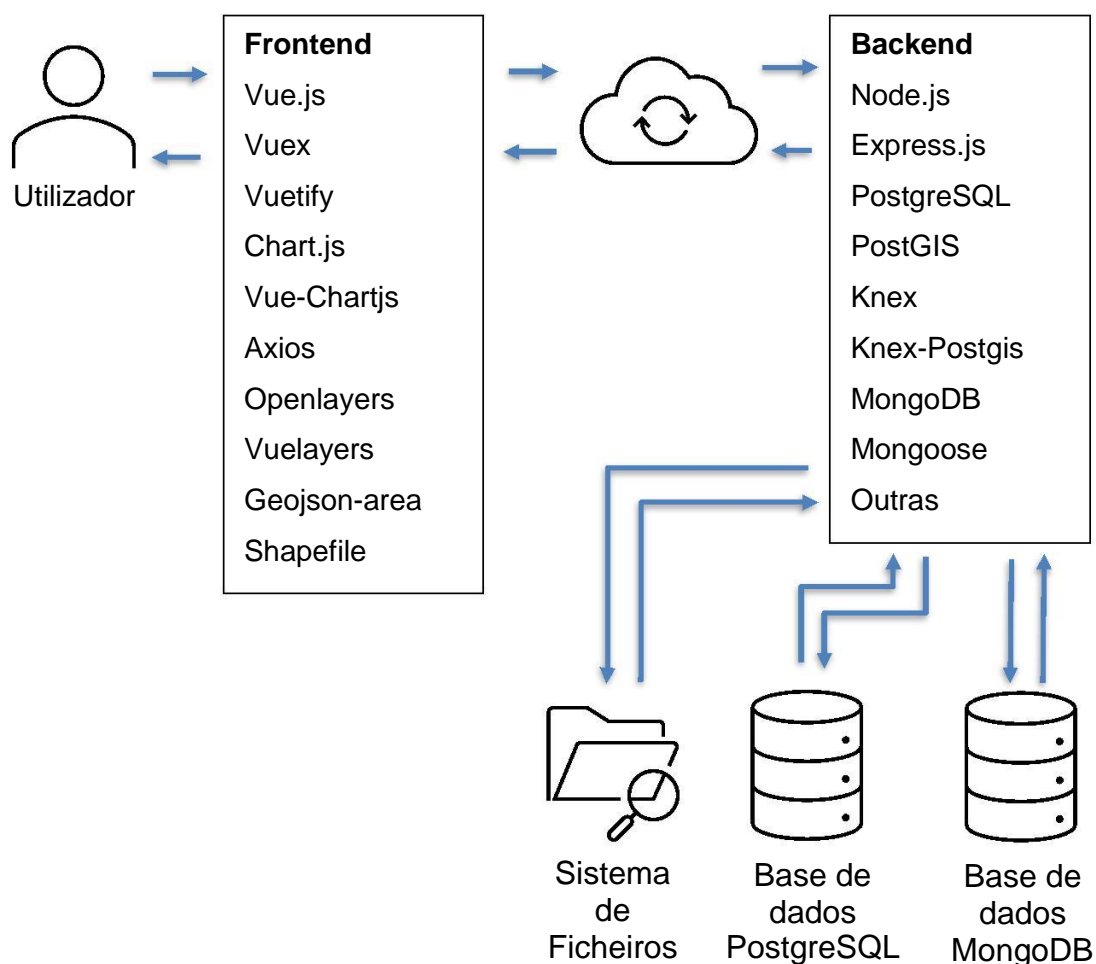


Figura 4. Arquitetura do sistema

3.4. As tecnologias

As tecnologias utilizadas para a realização deste projeto foram seleccionadas tendo em conta os critérios expostos no ponto anterior e são descritas nos pontos seguintes, distinguindo as que foram utilizadas no desenvolvimento do lado do servidor (*Backend*) e do lado do cliente (*Frontend*).

Como ambiente de desenvolvimento, foi utilizado um computador com o sistema operativo Linux Ubuntu versão 20.4 LTS e um servidor na *cloud* também com o sistema operativo Linux Ubuntu versão 20.4 LTS, onde foi criado um serviço SSH com um par de chaves público-privada para facilitar a transferência de ficheiros entre máquinas sem ter que introduzir passwords constantemente.

O desenvolvimento foi efetuado maioritariamente no computador local, sendo o servidor o destino final, onde a plataforma foi colocada em produção. No entanto, como o desenvolvimento da plataforma foi dividido entre o *Backend* e o *Frontend*, e de forma a possibilitar uma testagem contínua, quando o *Backend* se encontrou numa fase de produção, foi posto já em execução no servidor, juntamente com as bases de dados, sendo o acesso do *Frontend* ao *Backend* a partir desse momento feito de forma remota.

Como ferramenta de desenvolvimento foi utilizado o editor de código Visual Studio Code na versão 1.61.1 e o browser Google Chrome na versão 95.

Para instalação de pacotes foi utilizado o NPM (Node Package Manager), que se trata de um gestor de pacotes para o Node.js apresentado no ponto 3.4.1 deste capítulo.

Foi também criado um repositório Git para controlo de versões, onde ficou registado todo o histórico do desenvolvimento da plataforma.

3.4.1. Backend

No desenvolvimento da componente do lado do servidor da plataforma foram integradas um conjunto alargado de bibliotecas para responder aos requisitos definidos, nomeadamente sistemas de gestão de bases de dados para o armazenamento da informação sobre os prédios, *frameworks* para a criação da estrutura de serviços Web baseada numa arquitetura REST (Representational State Transfer) e bibliotecas para o controlo de acesso.

O ambiente de execução para o backend da plataforma selecionado foi o Node.js por permitir que seja utilizada a mesma linguagem de programação, o Javascript, no desenvolvimento da componente do lado do servidor e do lado do cliente.

O Node.js é um ambiente de execução JavaScript assíncrono orientado a eventos, projetado para desenvolvimento de aplicações escaláveis de rede (OpenJS Foundation, 2021). É descrito nesta secção dedicada ao *Backend* pois é o responsável pelo *runtime*, ou seja, todos os processos do lado do servidor são executados no Node na fase de desenvolvimento e também no ambiente de produção, mas o Node.js também é utilizado no *Frontend*, ainda que, apenas para gerar o *build* final, criando ficheiros HTML, CSS e JavaScript que são utilizados a partir desse momento para colocar em produção.

Trata-se de um software de código aberto, baseado no interpretador V8 do Google que permite executar código JavaScript sem a utilização de um browser (OpenJS Foundation, 2021). Esta potencialidade do Node.js faculta a possibilidade de escrever programas em linguagem JavaScript, que inicialmente estava associada ao desenvolvimento do lado do cliente, que será executado do lado do servidor. Como tal, o ambiente omite APIs (*Application Programming Interface*) JavaScript específicas do navegador e adiciona suporte para APIs de sistemas operativos mais tradicionais, incluindo *frameworks* de sistemas HTTP e arquivos (MDN Web Docs, 2021).

Para a implementação das funcionalidades da plataforma, foi necessário recorrer à utilização de várias *frameworks* que correm no ambiente Node.js, sendo o processo de instalação realizado utilizando o Node Package Manager (NPM) através da interface da linha de comandos.

Do ponto de vista do desenvolvimento de um servidor web, o Node possui vários benefícios (MDN Web Docs, 2021), nomeadamente:

- Desempenho excelente. O Node foi projetado para otimizar a taxa de transferência e a escalabilidade em aplicações web. É uma ótima ferramenta para resolver muitos problemas comuns no desenvolvimento da web (por exemplo, aplicações em tempo real).
- O código é escrito em "JavaScript simples e antigo". Isso significa menos tempo gasto para lidar com mudanças de código entre navegador e servidor web, não sendo necessária uma mudança na linguagem.
- JavaScript é uma linguagem de programação relativamente nova e apresenta algumas vantagens quando comparada com outras linguagens

utilizadas na programação da componente servidor (por exemplo Python, PHP, etc.). Muitas outras linguagens novas e populares compilam/convertem para JavaScript, permitindo que usemos também essas linguagens, como TypeScript, CoffeeScript, ClosureScript, Scala, LiveScript, etc.

- || O Gestor de Pacotes do Node (NPM, na sigla em inglês) dá acesso a centenas de milhares de pacotes reutilizáveis. O NPM possui a melhor coleção de dependências e também pode ser usado para automatizar a maior parte da cadeia de ferramentas de compilação.
- || É multiplataforma, com versões para diferentes sistemas operativos, como Microsoft Windows, OS X, Linux, Solaris, FreeBSD, OpenBSD, WebOS e NonStop. Além disso, tem excelente suporte de muitos fornecedores de alojamento na web, que muitas vezes fornecem documentação e infraestrutura específica para hospedar sites desenvolvidos em Node.
- || Possui uma comunidade de utilizadores muito ativa e disponível

Dado que o Node.js não inclui, optou-se pelo Express.js como tecnologia complementar para a implementação de funcionalidades/recursos de aplicações web, por exemplo, se quisermos que a aplicação possua diferentes métodos HTTP (GET, POST, DELETE, etc.), que faça a gestão de requisições de diferentes URL's ("rotas") ou apresente arquivos estáticos. Assim, foi necessário selecionar uma *framework* que efetuasse a gestão desta componente no desenvolvimento da plataforma, tendo-se optado pela *framework* web express.js. Esta opção deveu-se a se ter verificado que é uma das *frameworks* mais utilizadas com quase 20 milhões de downloads por semana³

Express.js é uma *framework* do Node que oferece soluções para:

- || Gerir requisições de diferentes métodos HTTP em diferentes URL's.

³ Número em constante crescimento. Fonte: <https://www.npmjs.com/package/express>

- ▮ Definir as configurações comuns da aplicação web, como a porta a ser utilizada para a conexão e a localização dos modelos que são usados para renderizar a resposta.
- ▮ Adicionar novos processos de requisição por meio de “middleware” em qualquer ponto da “fila” de requisições.

O Express é bastante minimalista, no entanto, os desenvolvedores têm liberdade para criar pacotes de middleware específicos com o objetivo de resolver problemas específicos que surgem no desenvolvimento de uma aplicação. Há bibliotecas para trabalhar com cookies, sessões, login de utilizadores, parâmetros de URL, dados em requisições POST, cabeçalho de segurança e muitos outros (MDN Web Docs, 2021).

Para o armazenamento e a gestão dos dados sobre os prédios, escolheu-se o sistema de gestão de base de dados PostgreSQL com a extensão PostGIS, por ser o sistema de código aberto com maior sucesso e mais avançado (Worsley & Drake, 2002).

O PostgreSQL é um sistema de gestão de base de dados relacional de código aberto que usa e estende a linguagem SQL combinada com muitos recursos que armazenam e escalam de forma segura as cargas de trabalho de dados mais complicadas (The PostgreSQL Global Development Group, 2021).

O PostgreSQL conquistou uma forte reputação pela sua arquitetura comprovada, confiabilidade, integridade de dados, conjunto de recursos robustos, extensibilidade e dedicação da comunidade de código aberto que suporta o software e que permite conseguir a constante disponibilização de soluções inovadoras.

PostgreSQL é executado em todos os principais sistemas operativos, é compatível com ACID (Atomicity, Consistency, Isolation and Durability) e foi desenhado de forma a poder ser facilmente complementado com novas capacidades / funcionalidades com base em extensões que podem ser carregadas e utilizadas como funcionalidades de base do motor da base de dados, como o caso da extensão PostGIS (The PostgreSQL Global Development Group, 2021).

PostGIS é uma extensão de base de dados espacial para a base de dados relacional PostgreSQL. Adiciona suporte para objetos geográficos, permitindo que

consultas de localização ou referentes a geometrias sejam executadas em SQL (PostGIS Project Steering Committee, 2021).

No desenvolvimento da plataforma SICAP, a base de dados PostgreSQL foi utilizada como a base de dados relacional para o armazenamento de todos os dados alfanuméricos assim como os dados geográficos associados às propriedades.

Para a construção dos comandos SQL e a comunicação com a base de dados, optou-se pela biblioteca open source Knex.js, que é descrito como um *query builder* para PostgreSQL, SQL Server, Mysql, MariaDB, Sqlite3, Oracle e Amazon Redshift, projetado para ser flexível, portátil e fácil de usar. Utiliza tanto o tradicional estilo de *Callbacks* do Node como as interfaces de *Promises* para um controlo de fluxo assíncrono mais limpo (knexjs.org, 2021).

O knex.js unifica a forma de fazer *queries* à base de dados por meio de linguagem JavaScript e como tem suporte completo para os diversos sistemas de base de dados, não dependendo de nenhum em específico, possibilita que seja facilmente exequível a troca de sistema de base de dados quando necessário, sem requerer adaptar ou alterar o código fonte da plataforma.

Foi ainda utilizada uma base de dados MongoDB, que é um sistema de base de dados orientado a documentos, de código aberto e multiplataforma. É classificado como um sistema de base de dados *NoSQL* não relacional e usa documentos semelhantes a JSON com esquemas. Não tem conceitos de tabelas, linhas ou SQL. Não há transações, conformidade com ACID, *joins*, chaves estrangeiras nem vários outros recursos chamados “padrão” (Hows, Membrey, & Plugge, 2019). É uma base de dados distribuída no seu núcleo, com alta disponibilidade e facilmente escalável horizontalmente.

Para facilitar a utilização da base de dados MongoDB, optou-se por recorrer à biblioteca Mongoose, que é uma ferramenta de modelação de objetos MongoDB projetada para funcionar num ambiente assíncrono. Inclui casting de tipo, validação e query Building. Uma das vantagens de utilizar o Mongoose é o de fornecer uma camada de abstração sobre o MongoDB.

Tal como o knex, o Mongoose oferece suporte para *Promises* e *Callbacks*.

Para implementação de alguns módulos da componente do lado do servidor da plataforma, foi necessário ainda integrar outras bibliotecas, nomeadamente:

- ┌ Para controlo de acesso à plataforma, aplicando um método criptográfico do tipo *hash* para codificar as palavras-passe, recorreu-se à biblioteca Bcrypt.js.
- ┌ Para controlar o acesso aos endpoints RESTful sem a criação do conceito de sessões, utilizou-se as bibliotecas Passport e Passport-jwt, que permitem a autenticação aplicando a estratégia de JSON web token, que é um padrão da Internet para a criação de dados com assinatura opcional e/ou criptografia cujo *payload* contém um JSON que transmite, de forma segura, informação entre as partes. Esse é criado através da biblioteca Jsonwebtoken, onde os tokens criados são assinados digitalmente usando um segredo privado ou uma chave pública/privada.
- ┌ Para facilitar o desenvolvimento da aplicação com separação lógica de arquivos e carregamento automático de scripts foi utilizada a biblioteca Consign.
- ┌ Para habilitar e configurar o *Cross-Origin Resource Sharing*, pois o que chamamos de Backend e Frontend da aplicação, são no fundo duas aplicações independentes que irão comunicar entre si, recorreu-se à biblioteca Cors
- ┌ Para carregar variáveis de ambiente (*environment*) de um arquivo .env em process.env utilizou-se a biblioteca Dotenv. O armazenamento da configuração do *environment* separado do código é baseado na metodologia Twelve-Factor App.
- ┌ Para validar e manipular datas fez-se uso da biblioteca Moment.
- ┌ Para lidar com multipart / form-data, usado para fazer upload de arquivos, recorreu-se à biblioteca Multer.
- ┌ Para agendar tarefas em datas específicas foi utilizada a biblioteca node-Schedule. Foi utilizada para recolher dados da base de dados de uma forma regular.
- ┌ Para reiniciar automaticamente a aplicação quando deteta mudanças nos arquivos foi utilizada a biblioteca Nodemon. Esta ferramenta foi utilizada apenas no desenvolvimento da aplicação.

3.4.2. Frontend

No desenvolvimento da componente do lado do cliente, na seleção das tecnologias, procurou-se selecionar *frameworks* e bibliotecas que facilitassem a construção e disponibilização de uma interface gráfica simples de utilizar para o utilizador, baseada em abordagens recentes e inovadoras.

Para a criação de uma camada de visualização com uma interface interativa e reativa, foi escolhida a biblioteca Vue, que é uma estrutura progressiva para construir interfaces de utilizador. A biblioteca concentra-se apenas na camada de visualização, mas é perfeitamente capaz de criar aplicações sofisticadas de página única quando usado em combinação com ferramentas modernas e *frameworks* de suporte (You, 2021).

Foi mencionado que a interface deve ser reativa. Para isso deve ser capaz de observar alterações no estado da aplicação e dos seus componentes, propagar notificações de alterações através da aplicação, apresentar (renovar) vistas automaticamente em resposta às alterações do estado e fornecer feedback oportuno às interações do utilizador (Hanchett & Listwon, 2018).

Existem alternativas ao Vue, havendo duas que se destacam das demais pela sua popularidade, o Angular e o React (Figura 5 e Figura 6).

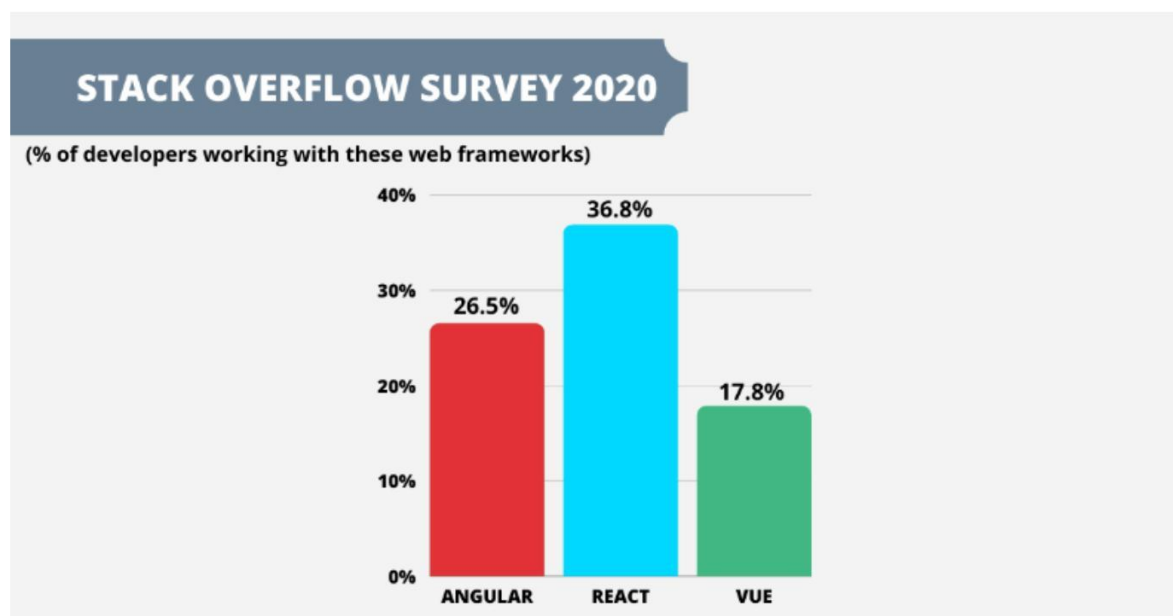


Figura 5. Popularidade das frameworks Angular, React e Vue

Na fase de pré-desenvolvimento foi tomada a decisão de utilizar o Vue, mas não sem primeiro fazer um comparativo das vantagens e desvantagens de cada uma das *frameworks*.

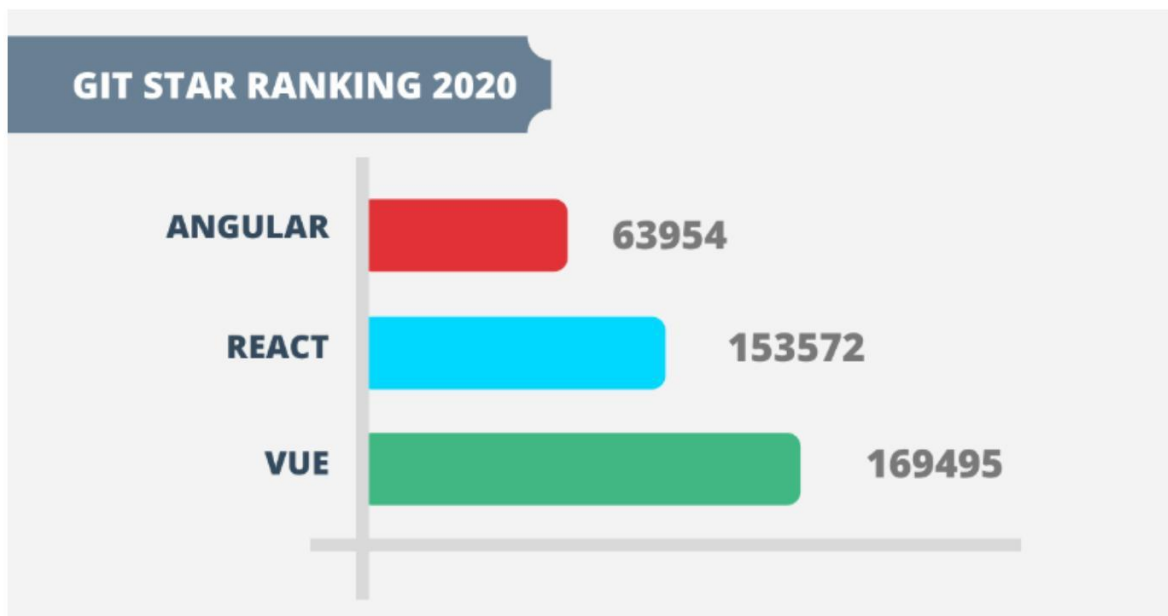


Figura 6. Classificação das frameworks Angular, React e Vue segundo o Git Star Ranking

São muitos os atributos de cada uma das *frameworks*. Seria necessário criar uma tabela bastante extensa para os enumerar a todos, por isso, destacam-se na Tabela 2 apenas algumas das características que pesaram na tomada da decisão:

Tabela 2. Comparativo entre as frameworks Angular, React e Vue

	Angular	React	Vue
Linguagem de programação	Typescript	Javascript	Javascript
Curva de aprendizagem	Alta	Moderada	Moderada
UI baseada em componentes	✓	✓	✓
Gestão de estado	✓	✗	✓
Routing	✓	✗	✓

Era pretendida uma *framework* para criar a camada de visualização que fosse baseada em componentes, utilizando uma linguagem de programação conhecida

como o Javascript e cuja curva de aprendizagem não fosse alta. A simplicidade da utilização, a possibilidade de gestão de estados e routing sem necessidade de bibliotecas externas, pesou na decisão entre o React e o Vue.

Quando é referido que o Vue trata internamente da gestão de estados é preciso que referir que o faz através do Vuex, que é um padrão de gestão de estado + biblioteca para aplicações Vue.js. O Vuex possibilita o armazenamento centralizado do estado de todos os componentes de uma aplicação numa *Store*, com regras que garantem que o estado só pode ser alterado de maneira previsível, através de *actions* e que depois de alterado (mutated) notifica as views que foi alterado (Koutifaris, 2018).

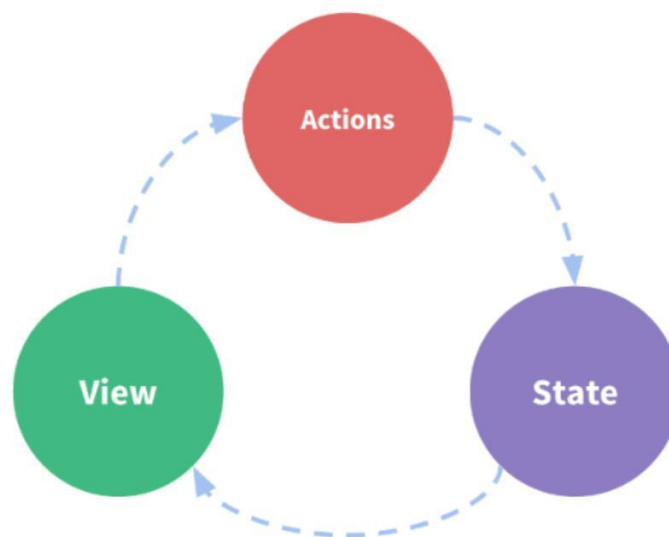


Figura 7. Vuex: Fluxo de dados unidirecional⁴

O Vuex também se integra com a extensão *devtools* oficial do Vue (abre uma nova janela) para fornecer recursos avançados, como depuração em tempo real e exportação / importação de estado (Vuejs.org, 2021).

Para a criação da interface de utilizador da plataforma, foi utilizada a biblioteca Vuetify, que é uma estrutura de UI completa construída sobre o Vue.js, com o

⁴ Fonte da figura: <https://vuex.vuejs.org/ptbr/>

objetivo de fornecer aos programadores as ferramentas que precisam para criar experiências de utilizador ricas e envolventes.

Ao contrário de outras *frameworks*, o Vuetify foi projetado desde o início para ser fácil de aprender e vantajoso de dominar, com centenas de componentes cuidadosamente elaborados a partir da especificação do Material Design, com uma abordagem de design *mobile-first*, o que significa que a aplicação funciona seja num telefone, tablet ou computador desktop (Leider & Leider, 2021).

Ainda com o objetivo de criar experiências ricas de utilizador, foram criados gráficos com o Chart.js, que é uma biblioteca JavaScript de código aberto gratuita para visualização de dados através de oito tipos de gráficos diferentes que suporta: barra, linha, área, pizza, bolha, radar, polar e dispersão (da Rocha, 2019).

O Vue-chartjs permite utilizar o Chart.js dentro do Vue, abstraindo a lógica básica, mas expondo o objeto Chart.js, oferecendo a máxima flexibilidade (Juszczak, 2021).

Para a criação de mapas interativos foi utilizada a biblioteca Openlayers, que facilita a inserção de um mapa dinâmico em qualquer página web, exibindo blocos de mapas, dados vetoriais e marcadores, carregados de qualquer fonte. O Openlayers foi desenvolvido para promover o uso de informações geográficas de todos os tipos (Openlayers.org, 2021). Para trazer a API do Openlayers para o mundo reativo do Vue.js, foi utilizada a biblioteca de componentes VueLayers, permitindo exibir mapas com camadas ladrilhadas, raster ou vetoriais, carregadas de fontes diferentes (Vershinin, 2021).

Foram ainda utilizadas duas pequenas bibliotecas, uma de nome Geojson-area para calcular a área de uma geometria GeoJSON e outra com o nome de Shapefile para fazer o *parser* dos ficheiros com o formato de arquivo com o mesmo nome que a biblioteca, ou seja, Shapefile. Trata-se de ficheiros que contêm dados geoespaciais em forma de vetor, usado por sistemas de informação geográfica.

Por último, para gerir as requisições e respostas HTTP entre o cliente e servidor, foi utilizada a biblioteca Axios, que é um cliente HTTP baseado em promessas para o Node.js e para o browser. É isomórfico, ou seja, pode ser executado no browser e no node.js com a mesma base de código. No lado do servidor usa o código nativo

do node.js, o módulo http, enquanto no lado do cliente (browser) usa XMLHttpRequests (Matt Zabriskie, 2021).

3.5. Níveis de acesso à plataforma

De acordo com os objetivos definidos para a plataforma, identificou-se dois tipos de utilizador: o utilizador padrão e o administrador. O tipo de utilizador é definido no ato do registo e só os administradores podem criar outros administradores.

O controlo dos níveis de acesso dos dois tipos de utilizador durante a execução do programa, é realizado aquando da invocação dos serviços à API (backend). Em cada uma dessas requisições é verificado se essa ação só pode ser efetuada por um administrador, ou se pode ser executada por um utilizador padrão. Caso só possa ser executada por um administrador o serviço da plataforma devolve um aviso.

Ao utilizador padrão apenas é permitida a consulta, enquanto o administrador pode efetuar todas as operações de criação, consulta, edição e remoção de propriedades.

3.6. Funcionalidades da plataforma

3.6.1. API Restfull

A plataforma conta com uma Interface de Programação de Aplicações (API, do inglês Application Programming Interface) do lado do servidor, que será responsável pelos serviços de requisição e resposta HTTP por parte do Frontend. Esta API assenta numa arquitectura REST, que fornece diretrizes para sistemas distribuídos comunicarem diretamente entre si, usando os princípios e protocolos web existentes, criando serviços web e APIs, sem a necessidade do protocolo SOAP ou quaisquer outros protocolos mais complexos. É uma arquitetura simples e fornece acesso aos recursos para que o cliente REST aceda e renderize os recursos no lado do cliente. A arquitetura REST usa vários formatos para representação de estados, como XML, JSON, Texto, imagens e assim por diante (Subramanian & Raj, 2019). Na plataforma SICAP, todas as requisições e respostas são expressas no formato JSON.

O objetivo da API é de simplificar as funções de criar, consultar, editar ou remover o conjunto de dados em que se baseia a plataforma, sendo eles os dados sobre os utilizadores, os proprietários, as propriedades com informações do registo e matriz, os distritos, concelhos e freguesias, as fotografias e os documentos.

3.6.2. Autenticação

Para poder utilizar a plataforma, o utilizador terá de efetuar a sua autenticação através da introdução de um email e palavra-passe. Esta ação cria uma requisição que é enviada para a API onde irá ser validado o login.

Antes de ser enviada a requisição, é feita uma primeira validação do lado do cliente. Chegada ao lado do servidor, é sujeita a nova validação. Caso o *payload* recebido não contenha algum dos campos email ou password é devolvido um alerta.

De seguida, é feita uma consulta à base de dados comparando palavras-passe utilizando a biblioteca Bcrypt.js pois a palavra-passe que está guardada na base de dados encontra-se encriptada. Caso a comparação seja bem-sucedida, é enviada uma resposta ao cliente com informações sobre o utilizador que acabou de se autenticar e incluída nessa resposta, também um *token* criado com a biblioteca *jsonwebtoken*.

O *token* é criado utilizando um segredo, uma palavra-chave guardada no ficheiro .env como uma variável de ambiente. Esta palavra-chave deve ser bem guardada em segredo pois a sua divulgação põe em causa a segurança de todo o sistema.

A partir deste momento qualquer autenticação é feita através do *token* e não através do email e password. Todas as requisições do cliente ao servidor irão incluir o *token* no cabeçalho da requisição HTTP e esse mesmo token será validado do lado do servidor utilizando a mesma biblioteca *jsonwebtoken*. A validade será feita através do segredo (palavra-chave) e pela verificação do prazo de expiração, que quando excedido, requererá nova autenticação por parte do utilizador.

3.6.3. Gestão de Utilizadores

A criação de novos utilizadores pode ser efetuada por um utilizador que seja administrador do sistema, para isso apenas tem que preencher o formulário com o nome, email e password do novo utilizador.

A password é introduzida duas vezes para efeitos de confirmação.

No ato da criação, é também definido se o novo utilizador irá ser administrador do sistema ou utilizador padrão. Os níveis de acesso à plataforma por parte do administrador ou do utilizador padrão podem definir-se segundo a Tabela 3:

Tabela 3. Níveis de acesso dos utilizadores

	Administrador	Utilizador Padrão
Utilizadores	É permitida a criação, visualização, edição e remoção de utilizadores	É negado o acesso a esta área
Camadas (Layers)	É permitida a adição e remoção de camadas	É negado o acesso a esta área
Proprietários	É permitida a criação, visualização, edição e remoção de proprietários	Apenas é permitida a consulta
Propriedades	É permitida a criação, visualização, edição e remoção de propriedades	Apenas é permitida a consulta
Mapa	Permitida a interação, podendo adicionar (ligar) camadas e utilizar filtros de pesquisa	Permitida a interação, podendo adicionar (ligar) camadas e utilizar filtros de pesquisa
Análise	Permitida a consulta	Permitida a consulta

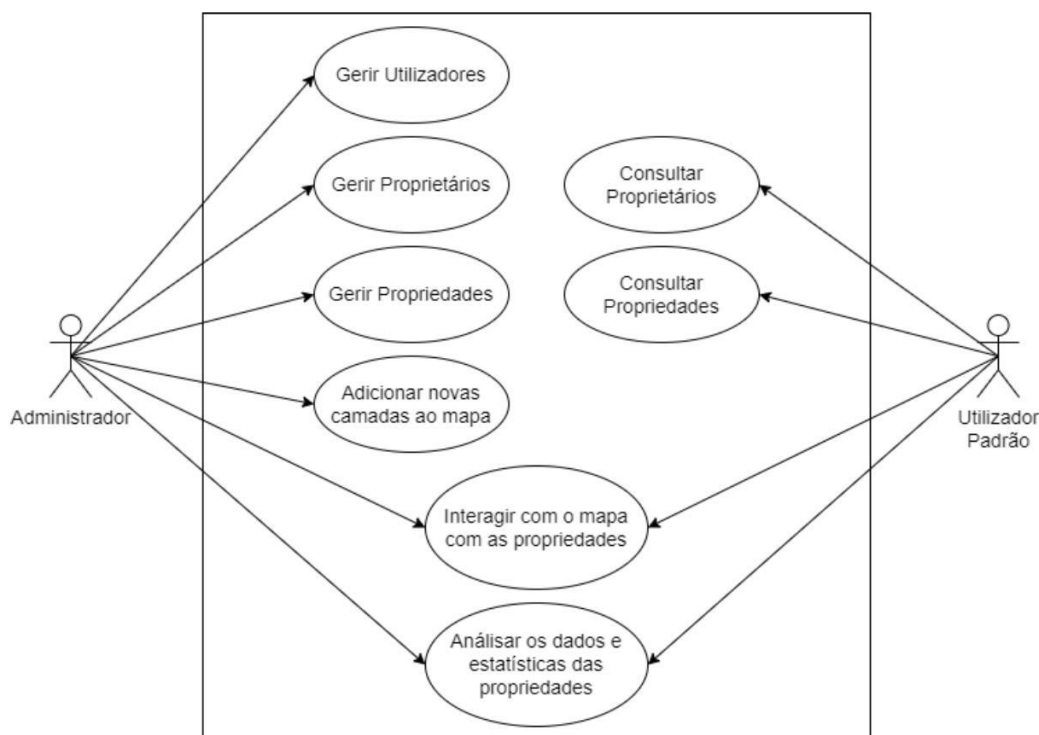


Figura 8. Casos de uso

A plataforma tem um “super administrador” que não aparece visível na listagem de utilizadores e cuja conta não é possível apagar. Quando não existirem outros utilizadores, será este o responsável por criar as primeiras contas.

A informação sobre os utilizadores é guardada numa tabela da base de dados de nome “users” (Figura 9), que é a única tabela que não está presente no modelo de dados da Figura 3 do capítulo 3.2.

```

sicap=# \d users
Table "public.users"
 Column |          Type          | Collation | Nullable |          Default
-----|-----|-----|-----|-----
 id     | integer               |           | not null | nextval('users_id_seq'::regclass)
 name   | character varying(255) |           | not null |
 email  | character varying(255) |           | not null |
 password | character varying(255) |           | not null |
 admin  | boolean               |           | not null | false
 deletedAt | timestamp with time zone |           |           |
Indexes:
 "users_pkey" PRIMARY KEY, btree (id)
 "users_email_unique" UNIQUE CONSTRAINT, btree (email)

```

Figura 9. Tabela users

Para a remoção de utilizadores, foi adotada a estratégia de *soft delete*, que consiste em não apagar verdadeiramente o utilizador nem o seu registo na base de dados, apenas lhe é atribuído uma data no campo *deletedAt* e partir desse momento é considerado como removido. Esta estratégia é útil quando são registadas ações atribuídas ao utilizador no formato de *logs* e que mais tarde poderão ser consultadas.

3.6.4. Gestão de Proprietários

A criação de novos proprietários é feita preenchendo um formulário com os dados:

- Nome
- Número de cartão de cidadão
- NIF
- Morada
- Email

É possível também, a qualquer momento, editar ou remover um proprietário.

3.6.5. Gestão de Propriedades

A criação de um novo prédio ou propriedade é feita preenchendo um formulário dividido em seis secções:

- ▮ Proprietário
- ▮ Registo
- ▮ Matriz
- ▮ Localização
- ▮ Ficheiros
- ▮ Parcela

É possível também, a qualquer momento, editar ou remover uma propriedade.

3.6.6. Mapa

A plataforma permite consultar um mapa, criado utilizando as bibliotecas openlayers e vuelayers, onde se pode consultar visualmente representadas as propriedades introduzidas.

É possível também, adicionar camadas (layers) a este mapa. Essas camadas podem ser adicionadas na forma de um ficheiro com o formato KML ou importadas de um web service utilizando o formato WMS. No próximo capítulo, na secção da apresentação da plataforma será mais perceptível como funcionam as layers.

O mapa permite também a utilização de filtros, apresentando apenas as propriedades pretendidas.

3.6.7. Análise

É possível fazer uma análise das propriedades mediante diversos critérios e apresentar os resultados no formato de gráficos. Nesta secção temos gráficos que nos dão informação acerca de:

- ▮ Propriedades por proprietário
- ▮ Propriedades por concelho
- ▮ Propriedades por natureza
- ▮ Propriedades por área
- ▮ Propriedades por IMI
- ▮ Propriedades por valor Patrimonial

Estes gráficos são criados utilizando as bibliotecas Chart.js e Vue-chartjs.

Ainda dentro do contexto da análise, e pensando num trabalho futuro, foi criada uma rotina do lado do servidor, utilizando a biblioteca node-Schedule para recolher dados da base de dados PostgreSQL de uma forma regular (neste momento definida para ser executada uma vez por dia). Esta rotina guarda os dados recolhidos, que são bastante básicos nesta fase (resumem-se ao número de proprietários e ao número de propriedades introduzidas) e guarda-os na base de dados MongoDB.

Como esta funcionalidade foi pensada para um trabalho futuro, ainda está em aberto a possibilidade de vir a recolher outros dados, e nesse sentido, a MongoDB por ser uma base de dados não relacional sem esquema, permite-nos fazer as alterações necessárias na devida altura sem prejuízo dos dados recolhidos até lá. Além do motivo apresentado, foi utilizada uma base de dados diferente pois a base de dados PostgreSQL está fechada ao acesso autenticado dos utilizadores da plataforma visto que contém dados sensíveis e confidenciais, enquanto os dados da base de dados MongoDB podem vir a ser públicos e utilizados por alguma ferramenta de apoio à decisão, como por exemplo o Power BI, alheia à plataforma SICAP.

3.6.8. Painel de Navegação

Por último, para navegarmos através das várias secções da plataforma, temos o painel de navegação, que como o nome indica permite que facilmente saltemos de uma secção para outra, de uma forma intuitiva.

4. APRESENTAÇÃO E ANÁLISE DE RESULTADOS

4.1. A plataforma

4.1.1. Iniciar sessão

O acesso à plataforma é realizado através de uma página Web, onde o utilizador deve antes de mais autenticar-se através do formulário de Login, introduzindo o email e a password com que está registado (Figura 10).

Caso já tenhamos iniciado sessão na plataforma anteriormente utilizando o mesmo browser e o token ainda estiver dentro da validade, o Login não será pedido e seremos logo encaminhados para o painel de navegação.

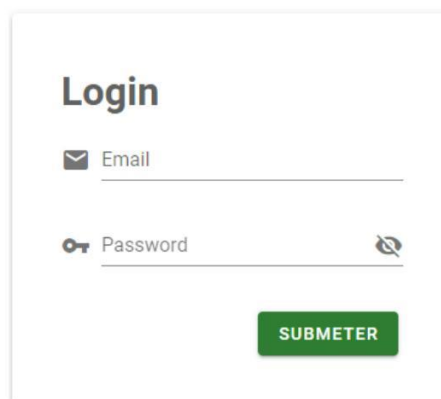
A screenshot of a login form titled "Login". It features two input fields: "Email" with an envelope icon and "Password" with a key icon and a toggle for visibility. A green "SUBMITER" button is positioned below the fields.

Figura 10. Formulário de Login

4.1.2. Painel de Navegação

Após ser efetuada a autenticação com sucesso, é apresentado o painel de navegação que permite percorrer as várias secções da plataforma de uma forma fácil e intuitiva (Figura 11).

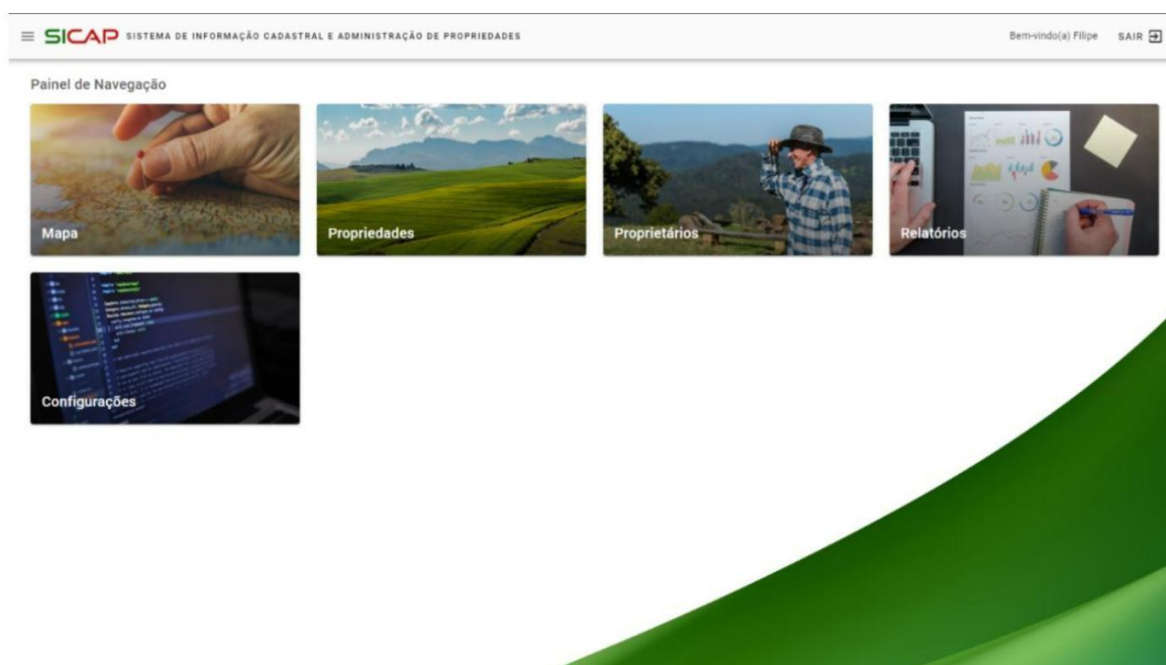


Figura 11. Painel de Navegação

As secções e subsecções existentes são as seguintes:

- || Mapa
- || Propriedades
- || Proprietários
- || Análise
- || Configurações:
 - || Utilizadores
 - || Layers
 - || Sobre

4.1.3. Configurações

4.1.3.1. Utilizadores

Na subsecção dos «Utilizadores», na listagem podemos ver os utilizadores registados, o seu nome, email e verificar se é administrador (Figura 12).

SICAP SISTEMA DE INFORMAÇÃO CADASTRAL E ADMINISTRAÇÃO DE PROPRIEDADES Bem-vindo(a) Filipe SAIR

Configurações

UTILIZADORES LAYERS SOBRE

Registo de utilizadores

Nome
Introduza o nome do utilizador



Email
Introduza o email do utilizador

Password
Introduza a password do utilizador

Confirmação de Password
Confirme a password do utilizador

Administrador

GUARDAR CANCELAR

Id	Nome	Email	Admin	Ação
1	Filipe Dias Rodrigues	filipedr@live.com	Sim	 

Utilizadores por página 5 1-1 of 1

Figura 12. Registo de Utilizadores

Podemos adicionar novos utilizadores, preenchendo o formulário com o nome, email, password e confirmação de password. Podemos também editar ou remover utilizadores existentes.

Esta subsecção só está disponível para administradores, sendo o seu acesso negado aos utilizadores padrão.

4.1.3.2. Layers

Nesta subsecção podemos adicionar ou remover *Layers* que serão apresentadas no mapa, conforme descrito no ponto 3.6.6 e no ponto 4.1.6.

São aceites ficheiros no formato KML, ou então URL's de *Web Services* no formato WMS.

Depois de inseridas as *layers*, quer seja em ficheiro ou em formato de URL, ficam guardadas na base de dados e estarão sempre disponíveis até serem apagadas (Figura 13).

Configurações

UTILIZADORES **LAYERS** SOBRE

Layer em formato KML

Layer Nome da Layer

Layer em formato WMS

URL Nome da Layer

Id	Nome	Path	Ação
55	CAOP	http://mapas.dgterritorio.pt/wms-inspire/caop/continente?service=WMS&REQUEST=GetCapabilities&VERSION=1.3.0	<input type="button" value="X"/>
56	PDM Monção	/uploads/layers/1634417945564_PDM_Moncao.kml	<input type="button" value="X"/>

Layers por página 5 1-2 of 2

Figura 13. Layers

4.1.3.3. Sobre

Esta subsecção contém apenas informações sobre a plataforma e sobre as pessoas envolvidas no projeto.

4.1.4. Proprietários

Na secção «Proprietários» podemos consultar a listagem com dados dos proprietários introduzidos. Os dados disponíveis são: Id, nome, número de cartão de cidadão, número de identificação fiscal, morada e email (Figura 14).

O utilizador padrão pode apenas consultar a listagem enquanto o utilizador administrador pode também criar, editar ou remover os proprietários existentes.

Proprietários

Nome
Introduza o nome do proprietário

Cartão de Cidadão
Introduza o n.º de cartão de cidadão

NIF
Introduza o NIF do proprietário

Morada
Introduza a morada do proprietário

Email
Introduza o email do proprietário

GUARDAR CANCELAR

Id	Nome	CC	NIF	Morada	Email	Ações
1	Fillpe Rodrigues	123456789	987654321	Av. da Liberdade n.º 74	fillpedr@live.com	 
2	Cátia Gil	102030405	908070605	Rua Direita n.º 35	catia_gil@hotmail.com	 

Proprietários por página 5 1-2 of 2 < >

Figura 14. Proprietários

4.1.5. Propriedades

A secção das «Propriedades» inclui as operações de administrador para inserir, consultar, editar ou remover propriedades. O utilizador padrão pode apenas consultar.

Serão apresentados primeiro os passos necessários para inserir uma propriedade, sendo a seguir demonstrado como cada propriedade pode ser consultada, como podemos ver as fotografias, e como podemos ver os seus documentos associados.

Aquando da introdução de uma nova propriedade, essa inserção é feita passo a passo, preenchendo formulários referentes a dados do proprietário, do registo, da matriz, da localização, de imagens e ficheiros e da parcela descritos nos pontos seguintes:

4.1.5.1. Proprietário

Neste passo escolhemos um proprietário entre os registados na base de dados no ponto 4.1.4. Essa escolha é feita através de uma caixa de escolha, sendo o preenchimento dos campos efetuado automaticamente mediante a escolha do proprietário (Figura 15).

Definido o proprietário, clicamos no botão «Continuar» para avançar para o seguinte passo. Este procedimento será utilizado para o preenchimento sucessivo de todos os elementos necessários à introdução de uma propriedade.

Registo de Propriedades

1 Proprietário — 2 Registo — 3 Matriz — 4 Localização — 5 Ficheiros — 6 Parcela

Proprietário

Nome do Proprietário
Filipe Rodrigues

Cartão de cidadão
123456789

NIF
987654321

Morada
Av. da Liberdade n.º 74

Email
filipedr@live.com

CANCELAR CONTINUAR

Figura 15. Registo de Propriedades: Proprietário

4.1.5.2. Registo

Neste passo, o utilizador deverá preencher os campos do formulário com os dados sobre a propriedade que normalmente se encontra na certidão do registo predial (Figura 16).

Registo de Propriedades

1 Proprietário — 2 Registo — 3 Matriz — 4 Localização — 5 Ficheiros — 6 Parcela

Certidão de Registo

Número de Registo
1051/20140122

Conservatória
Monção

Direitos (opcional)

Restrições (opcional)

Responsabilidades (opcional)

CANCELAR VOLTAR CONTINUAR

Figura 16. Registo de Propriedades: Registo

Alguns campos são de preenchimento obrigatório, como o número de registo e o nome da conservatória. Outros são de preenchimento facultativo, como os direitos, as restrições e as responsabilidades.

4.1.5.3. Matriz

O formulário seguinte, referente à «Matriz» deve ser preenchido com os dados que habitualmente se encontram na caderneta predial do serviço de finanças (Figura 17).

Todos os campos são de preenchimento obrigatório.

Os campos referentes ao distrito, concelho, e freguesia são automaticamente preenchidos, consultando primeiro na base dados a tabela dos distritos e depois filtrando e apresentando apenas os concelhos desse distrito. A mesma lógica é a seguir utilizada com as freguesias.

Registo de Propriedades

Proprietário Registo 3 Matriz 4 Localização 5 Ficheiros 6 Parcela

Caderneta Predial

Número de artigo na matriz: 552
Natureza: Urbano

Localização: Rua Sá da Bandeira

Distrito: Viana do Castelo
Concelho: Monção
Freguesia: União das freguesias de Monção e Troviscc

Confrontação Norte: Adriano Augusto Pinto Junior
Confrontação Sul: Rua Sá da Bandeira

Confrontação Nascente: Rua João de Pinho
Confrontação Poente: Proprietário

Valor Patrimonial: 51 670,53
Valor IMI: 130,32

CANCELAR VOLTAR CONTINUAR

Figura 17. Registo de Propriedades: Matriz

4.1.5.4. Localização

No passo referente à localização é apresentado um mapa interativo. O browser irá fazer uso dos serviços de localização (caso estejam autorizados pelo utilizador) e irá centrar o mapa na localização onde nos encontramos. Caso a propriedade que

pretendemos registar se encontrar noutro lugar, o mapa tem funções de arraste e zoom permitindo navegar no mapa até encontrarmos a localização pretendida.

Depois utilizaremos o cursor clicando nos pontos das extremidades da propriedade sobre o mapa para assim a delinear. Desta forma iremos desenhar uma geometria correspondente à área da propriedade (Figura 18).

O desenho que acabamos de fazer pode ser editado, apagado e refeito as vezes que forem necessárias.

É possível alternar entre três vistas possíveis do mapa, ou melhor, observar o mapa utilizando três serviços diferentes:

- Open Street Maps
- Bing Maps (Vista Aerial)
- Mapas da Direção Geral do Território

Qualquer um destes mapas será apresentado segundo o sistema de cartografia ESPG:4326.

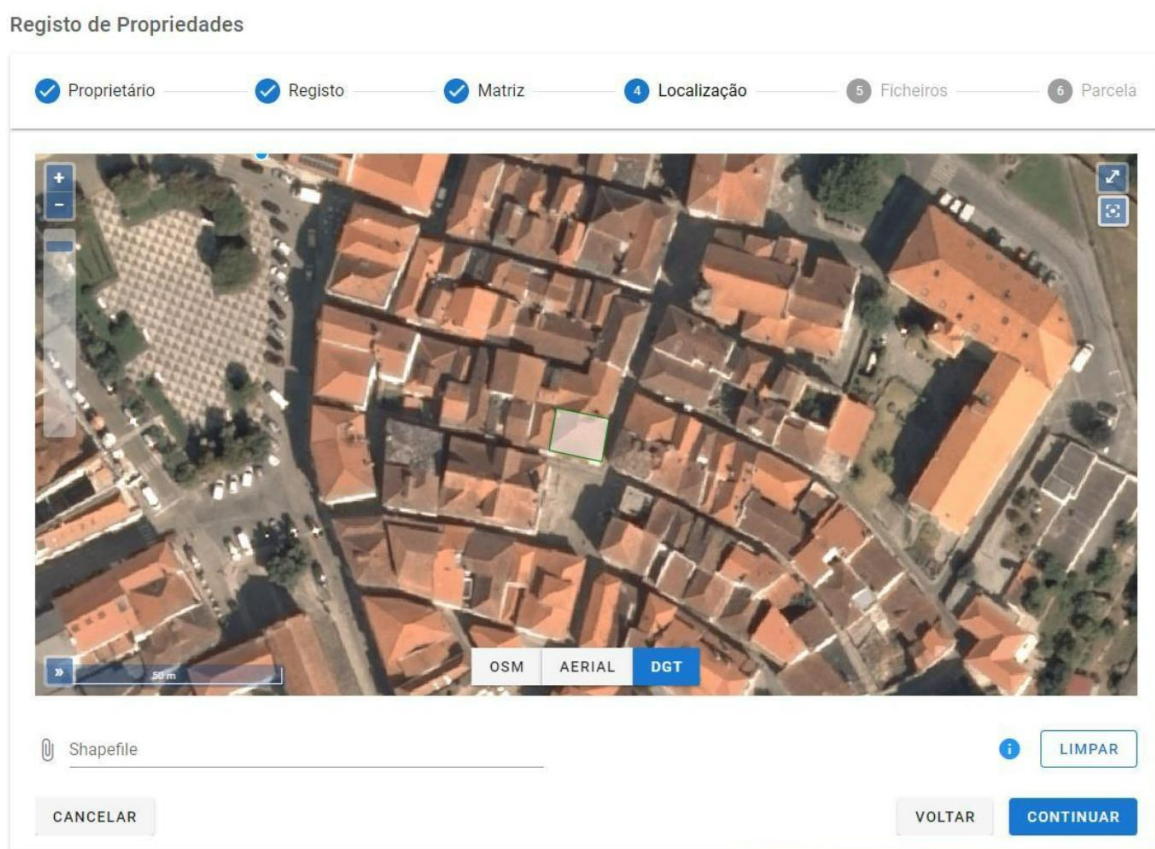


Figura 18. Registo de Propriedades: Localização

No sentido de permitir a integração da plataforma com outros métodos de recolha de dados no terreno, foi incluída uma outra alternativa que se baseia na importação de um ficheiro no formato Shapefile.

Este tipo de ficheiros pode ser obtido, por exemplo, através de um levantamento no local utilizando aparelhos específicos para esta função que normalmente já usam este tipo de ficheiros para armazenar os dados recolhidos.

O processo de identificação dos limites geográficos da propriedade, quando é registada esta informação, inclui, além de guardar a geometria (polígono), utilizar uma função do POSTGIS para guardar um ponto central (ST_PointOnSurface) que será doravante utilizado como ponto de referência (localização) para a propriedade. A opção pela função ST_PointOnSurface deveu-se a esta garantir que o ponto determinado se encontra no interior da área da propriedade (PostGIS Project Steering Committee, 2021). Quando voltarmos a abrir esta propriedade, no caso de a querermos editar, por exemplo, o mapa já será centrado usando esse ponto de referência.

4.1.5.5. Ficheiros

Para melhor identificarmos uma propriedade numa listagem, a plataforma permite atribuir uma imagem de capa, uma imagem que facilmente identifique a propriedade. Essa imagem de capa pode ser introduzida nesta secção (Figura 19).

Além da imagem de capa podem também ser adicionados à propriedade ficheiros como documentos ou fotografias.


Por exemplo, no caso de uma casa, poderíamos colocar uma fotografia da vista exterior como imagem de capa e depois acrescentarmos as fotografias que quiséssemos do interior à área das fotografias. Como documentos podíamos acrescentar uma cópia da caderneta predial ou da certidão de registo.

Registo de Propriedades

Proprietário ✓ Registo ✓ Matriz ✓ Localização ✓ Ficheiros ✓ 6 Parcela

Ficheiros

Imagem de capa



Fotografias
0 files (0 B in total)

Ver	Fotografias	Apagar
	terraço.jpg	

Fotografias por página 5 1-1 of 1 < >

Documentos
0 files (0 B in total)

Ver	Documentos	Apagar
	Vale do lobo - Caderneta predial.pdf	
	CertidaoPermanente-PP-1913-19040-160417-001051.pdf	

Documentos por página 5 1-2 of 2 < >

CANCELAR VOLTAR CONTINUAR

Figura 19. Registo de Propriedades: Ficheiros

4.1.5.6. Parcela

Sendo um dos objetivos da plataforma a compatibilidade com o registo predial nacional, esta secção permite acrescentar o NIP (Número identificador do prédio). Como o cadastro nacional ainda não está concluído, ainda existem muitas propriedades sem NIP. Este campo, estava previsto ser de preenchimento obrigatório, mas dada a realidade do país, ficou como preenchimento facultativo (Figura 20).

Complementarmente podemos assinalar se se trata de uma fração autónoma, ou não, no caso de ser um apartamento pertencente a um edifício com várias frações, por exemplo.

Temos três tipos de área diferente que podemos introduzir:

- || Área da parcela – Este valor pode ser introduzido manualmente ou pode ser utilizada a função de calcular área através da geometria criada.
- || Área de implantação do edifício

- || Área bruta de construção – Por exemplo, se se tratar de um edifício com vários pisos, será igual à soma da área dos pisos

Registo de Propriedades

Registo de Propriedades

✓ Proprietário — ✓ Registo — ✓ Matriz — ✓ Localização — ✓ Ficheiros — 6 Parcela

Parcela

NIP Fração Autónoma

Área da parcela 84 Área de implantação do edifício 84 Área bruta de construção 288

Descrição
Prédio com 4 pisos em propriedade total com 2 divisões com utilização independente

Data de aquisição 2019-09-30 × Valor de aquisição 240 000,00 Data da avaliação Avaliação

Válido desde Válido até

Figura 20. Registo de Propriedades: Parcela

Na parte final existem vários campos de preenchimento facultativo, como:































- || Data de aquisição
- || Valor de aquisição
- || Data de avaliação
- || Avaliação
- || Válido desde
- || Válido até

Por último, vamos submeter, criando assim uma nova propriedade na base de dados.

4.1.5.7. Consultar Propriedades

Depois de submetida, podemos consultar a propriedade de várias formas, sendo a primeira, a própria listagem de propriedades da secção «Propriedades» (Figura 21).

Propriedades

Ver	Imagem	Descrição	Área(m2)	Freguesia	Concelho	Ações
  		Prédio com 4 pisos em propriedade total com 2 divisões com utilização independente	84	União das freguesias de Monção e Troviscoso	Monção	 
  		Prédio com 3 pisos em propriedade total sem andares nem divisões de utilização independente	1122	União das freguesias de Monção e Troviscoso	Monção	 
  		Terreno para construção	1271	Seixas	Caminha	 
  		Loja	87	União das freguesias de Mazedo e Cortes	Monção	 
  		Prédio urbano para habitação	694	União das freguesias de Monção e Troviscoso	Monção	 

Propriedades por página 5 1-5 of 5 < >

Figura 21. Listagem de Propriedades

Aqui podemos consultar um número reduzido de dados acerca da propriedade, mas a imagem de capa, principalmente, ajuda a identificar a propriedade que queremos consultar e assim podemos utilizar as outras opções de visualização que se encontram do lado esquerdo (Figura 22).

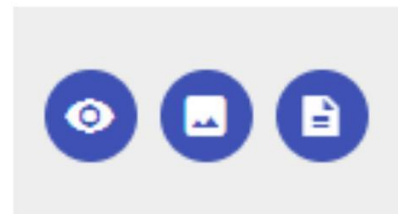


Figura 22. Botões de consulta

O primeiro ícone à esquerda permite que vejamos a ficha sobre a propriedade, onde tem a totalidade da informação alfanumérica, a fotografia de capa e um mapa com a localização (Figura 23).

Esta ficha foi composta de forma a facilitar a impressão caso se deseje guardar a mesma informação em formato papel.

É flexível no formato vertical, permitindo sempre acrescentar mais informação, o que faz com que por vezes não caiba tudo numa página apenas, como neste caso em que dividimos a informação entre a Figura 23 e a Figura 24.

**NIP:**

Descrição: Prédio com 4 pisos em propriedade total com 2 divisões com utilização independente

Área: 84 m²

**Registo**

Número de registo: 1051/20140122 **Conservatória:** Monção

Direitos: ---- **Restrições:** ---- **Responsabilidades:** ----

Proprietário

Nome do proprietário: Filipe Rodrigues

Morada do proprietário: Av. da Liberdade n.º 74

CC: 123456789 **NIF:** 987654321 **Email:** filipedr@live.com

Figura 23. Consultar Propriedade

Matriz

Número de artigo: 552 **Natureza:** Urbano

Localização: Rua Sá da Bandeira

Freguesia: União das freguesias de Monção e Troviscoso **Concelho:** Monção

Distrito: Viana do Castelo

Confrontações: Norte: Adriano Augusto Pinto Junior Sul: Rua Sá da Bandeira Nascente: Rua João de Pinho Poente: Proprietário

Valor patrimonial: 51670.53€ **Valor IMI:** 130.32€

Figura 24. Consultar Propriedade (continuação)

O ícone do centro (Figura 22) permite ver as fotografias associadas à propriedade (Figura 25). Estas fotografias são apresentadas em forma de carrossel.

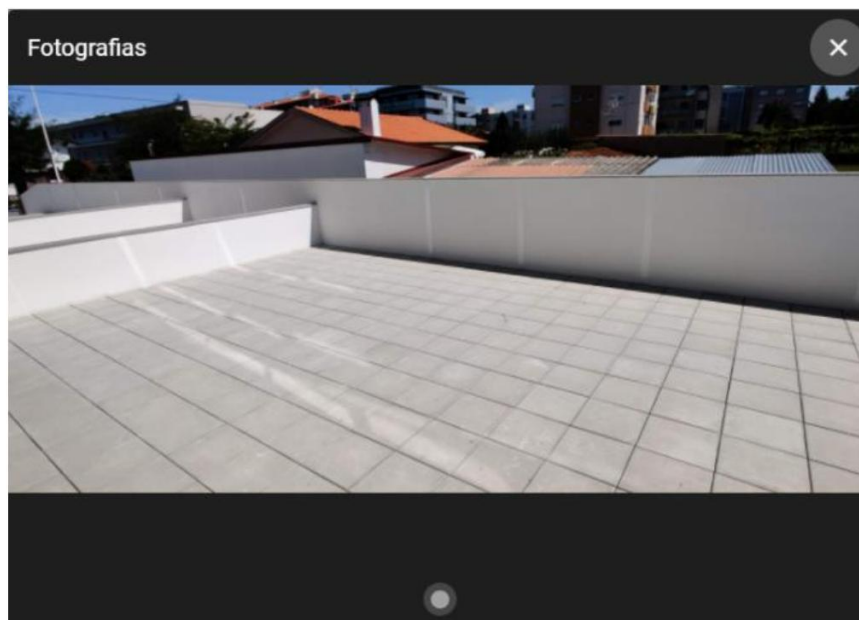


Figura 25. Consultar Propriedades: Fotografias

E por último, o ícone da direita (Figura 22) permite consultar os documentos associados à propriedade (Figura 26).

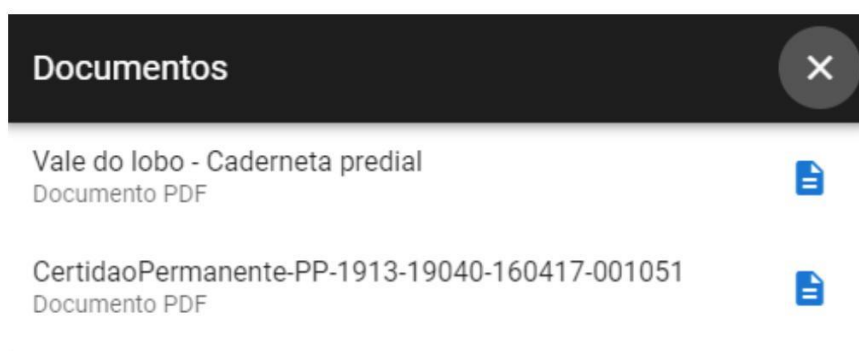


Figura 26. Consultar Propriedades: Documentos

4.1.6. Mapa

Na secção do «Mapa» podemos observar todas as propriedades introduzidas, podendo adicionar novas *layers* e utilizar filtros.

Podemos utilizar vários serviços web de mapas, tendo à disposição os mapas do Bing CanvasGrey, Bing Aerial, Open Street Maps e os mapas da Direção Geral do Território (Figura 27).



Figura 27 – Serviços de mapas

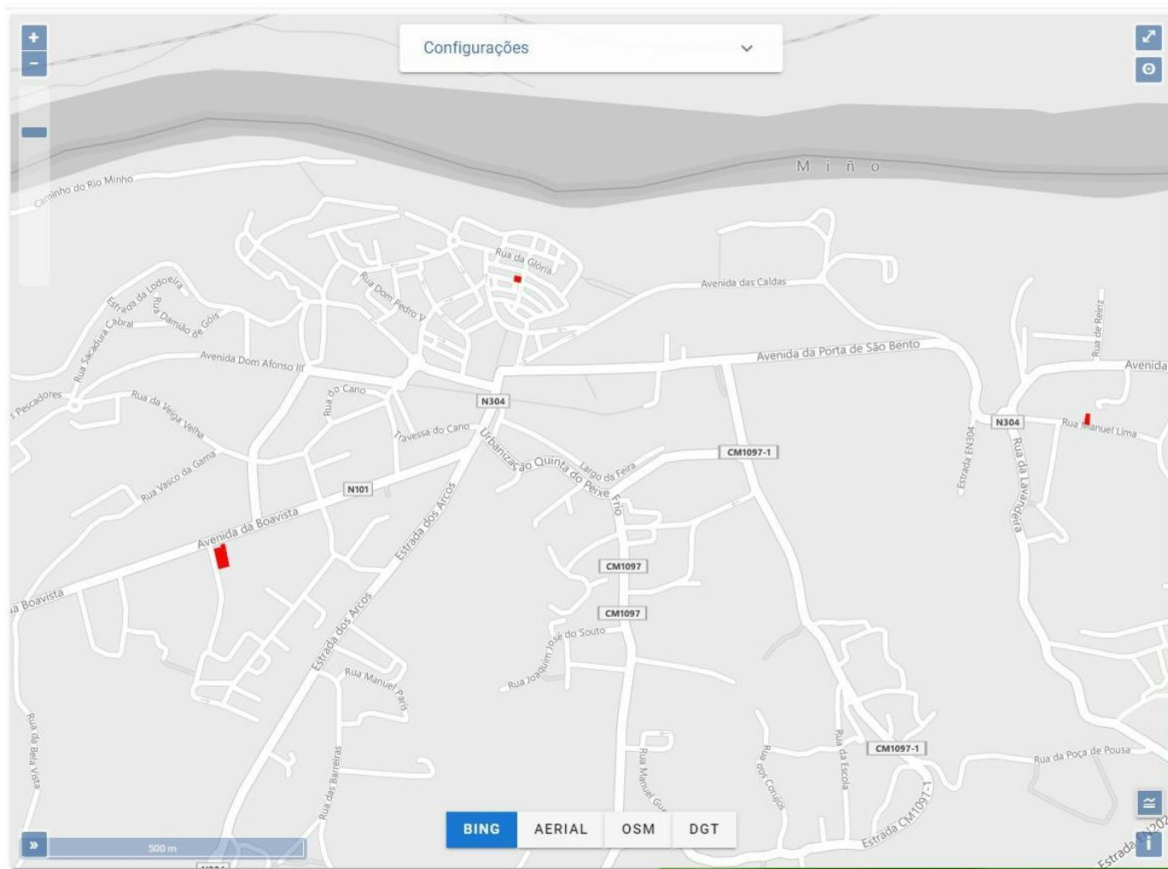


Figura 28. Mapa

No exemplo apresentado na Figura 28 podemos ver três propriedades a vermelho com opacidade a 100%, isto quer dizer que se trata de propriedades urbanas cujo valor patrimonial é superior a dez mil euros. Estas informações encontram-se na legenda (Figura 29) e para a abrir apenas temos que clicar no botão.

Na caixa das configurações (Figura 30) podemos acrescentar *layers*, o que fará com que sejamos reencaminhados para a subsecção das *layers* descrita em 4.1.3.2.

Natureza	Valor Patrimonial
Urbano	0 ≤ 100€
Misto	100 ≤ 1.000€
Rústico	1.000 ≤ 10.000€
	> 10.000€

Figura 29. Legendas Natureza / Valor

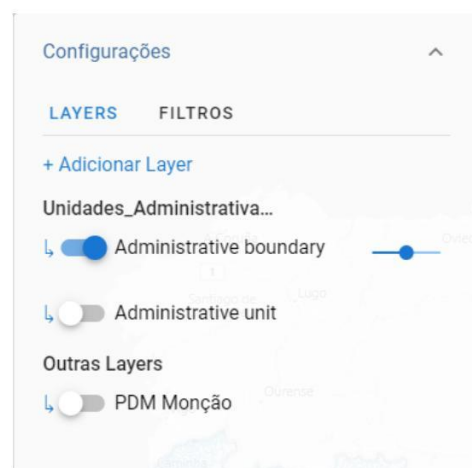


Figura 30. Layers

4.1.7. Análise

A plataforma permite-nos fazer uma análise das propriedades introduzidas mediante determinados critérios:

- ▮ Propriedades por proprietário (Figura 34)
- ▮ Propriedades por Concelho (Figura 34)
- ▮ Propriedades por natureza (Figura 34)
- ▮ Propriedades por área (Figura 35)
- ▮ Propriedades por IMI (Figura 35)
- ▮ Propriedades por valor patrimonial (Figura 36)

Para a apresentação dos resultados utilizamos uma série de gráficos com diferentes formatos.

Análise

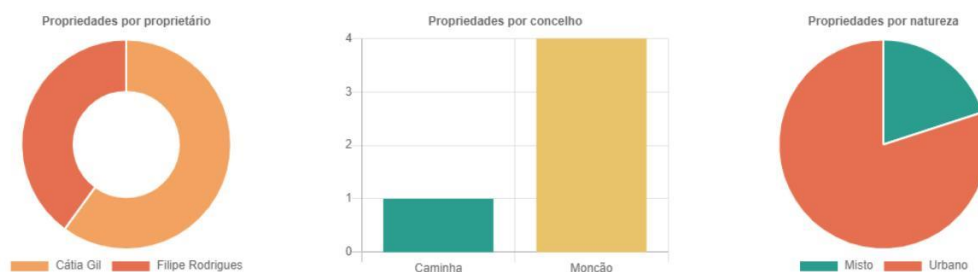


Figura 34. Análise: Propriedades por proprietário, concelho e natureza



Figura 35. Análise: Propriedades por área e por IMI

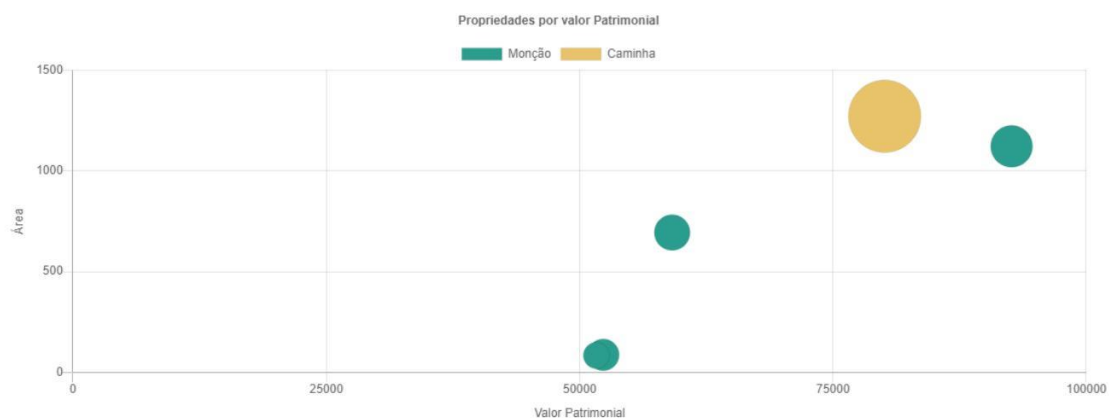


Figura 36. Análise: Propriedades por Valor Patrimonial

Neste último gráfico acabamos por conseguir representar quatro dimensões. Temos o valor patrimonial no eixo dos XX, a área no eixo dos YY, o concelho representado pela cor dos círculos e o valor do IMI pelo tamanho do círculo.

4.2. Testes

Durante todo o desenvolvimento da plataforma foram realizados testes nucleares e criadas funções de teste e validação, que de forma automática e contínua, testam os processos evitando erros indesejados.

Todas as requisições, antes de efetuadas são validadas do lado do cliente e depois de recebidas pelo servidor, voltadas a testar.

Um exemplo de função de validação do lado do servidor é a função *existsOrError*, utilizada na invocação dos serviços da API Restfull, que para a inserção ou atualização de dados recebe um *Payload*. Esse *Payload* contém dados com valores que são testados quanto à sua existência, ou seja, se não estão vazios. Para isso efetua diversas verificações uma vez que, em *javascript* existem várias condições em que um valor pode ser considerado inexistente, desde ter o valor *null*, ter um valor em branco ou *undefined*.

```
function existsOrError(value, msg) {
  if (!value) throw msg;
  if (Array.isArray(value) && value.length === 0) throw msg;
  if (typeof value === 'string' && !value.trim()) throw msg;
}
```

Segundo a mesma lógica foi criada uma função *notExistsOrError* e uma *equalsOrError*.

Estas funções são utilizadas na validação dos parâmetros recebidos, como é o caso da inserção de uma parcela nova em que alguns campos de preenchimento obrigatório são testados. Caso não passe no teste é devolvida um erro.

```
try {
  existsOrError(payload.numeroArtigo, 'Número de artigo não introduzido');
  existsOrError(payload.natureza, 'Natureza não introduzida');
  existsOrError(payload.localizacao, 'Localização não introduzida');
  existsOrError(payload.freguesia.dicofre, 'Freguesia não introduzida');
  existsOrError(payload.confrontacoes, 'Confrontações não introduzidas');
  existsOrError(payload.valorPatrimonial, 'Valor Patrim. não introduzido');
  existsOrError(payload.valorIMI, 'Valor do IMI não introduzido');
} catch (error) {
  return res.status(400).send(error);
}
```

Do lado do cliente, as requisições também só são feitas se obedecerem a um conjunto de regras pré-definidas. As regras de validação são implementadas recorrendo a uma estrutura e métodos do próprio Vue.js e Vuetify.js e apenas temos que as definir. No código fonte seguinte podemos ver um exemplo dos campos do formulário de Login que são compostos por endereço de email e password:

```
rules: {
  required: (value) => !!value || 'Required.',
  min: (v) => v.length >= 6 || 'Min 6 characters',
  email: (value) => {
    const pattern =
      /^(?!(^<>()[\]\|\.\.,;\s@"]+(\.[^<>()[\]\|\.\.,;\s@"]+)*)(("[.+"))@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.([a-zA-Z-0-9]+\.)+[a-zA-Z]{2,}))$)/;
    return pattern.test(value) || 'Invalid e-mail.';
  },
},
```

Os campos são de preenchimento obrigatório, a password tem que ter no mínimo seis caracteres e o email tem de obedecer à formatação padrão dos endereços de emails, o que aqui se verifica através de uma *regex*.

5. CONCLUSÕES E TRABALHO FUTURO

Neste projeto foi proposta uma plataforma para gestão cadastral e administração de propriedades, que permita registar dados alfanuméricos e geográficos, e possibilite a realização de operações simples de análise e visualização de métricas e estatísticas através dos dados das propriedades introduzidas.

A proposta focou-se em reunir informações necessárias e relevantes para a gestão de propriedades, a fim de criar um modelo de dados que considere a realidade portuguesa, mas também as orientações e standards europeus e internacionais.

Em primeiro lugar realizou-se uma revisão de literatura sobre o enquadramento, importância e desenvolvimento do cadastro nacional, sobre as especificações para os dados espaciais de parcelas e sobre os modelos de dados no domínio da administração de propriedades.

Um dos problemas identificados no cadastro de propriedades decorre do facto de cada prédio ter os seus dados dispersos entre as várias entidades competentes, o que dificulta a sua consulta.

Outro problema identificado, provavelmente o mais grave, foi a inexistência de cadastro de base geométrica e atualizado na grande maioria do território nacional e a falta de informação geográfica dos prédios.

Neste contexto pretendeu-se com a plataforma desenvolvida dar resposta às carências identificadas, reunindo as informações necessárias no mesmo sistema, possibilitando a inserção de objetos geográficos levantados de forma independente nos prédios sem cadastro e com a compatibilidade necessária para a importação dos mesmos objetos geográficos onde o cadastro já existe. Assim a plataforma permite a gestão dos dados de propriedades, incluindo o limite geográfico, os dados de registo nas finanças, da conservatória do registo predial e outros documentos associados e ainda foi tida em conta a inclusão de atributos de interesse e que normalmente não são mencionados nos documentos oficiais, como é o caso do valor do Imposto Municipal sobre Imóveis (IMI) que o proprietário deve pagar anualmente.

Para potenciar o uso da plataforma, os dados disponíveis podem ser utilizados para realizar diversas operações de análise e estatística, como por exemplo fazer uma

consulta para saber que propriedades têm uma área maior, quais as que tem um valor patrimonial mais alto, ou então, obter uma média do preço por metro quadrado numa determinada zona.

Como exemplo de aplicação prática, estudou-se o caso de uma entidade privada ou associação, que por ser proprietária de múltiplas parcelas de terreno em diferentes localizações necessita uma ferramenta para fazer a sua gestão.

Ao longo do trabalho concluiu-se que a plataforma cumpre com os objetivos propostos, mas tem um maior potencial, sendo, com o devido tempo possível aumentar as suas capacidades, acrescentando funcionalidades, como por exemplo, a localização automática aquando do preenchimento do campo relativo à localização, verificação de geometrias para identificar sobreposições, criar novos filtros e novos gráficos para análise.

Outra área de aplicabilidade do presente trabalho seria criar um website para acesso público, ficando a plataforma a funcionar como um BackOffice que permitisse, através das informações recolhidas na plataforma, criar um mercado fundiário online ou banco de terras, facultando uma estrutura para a compra, venda, arrendamento ou permuta de parcelas de utilização agrícola e florestal.

Uma funcionalidade já implementada a pensar num trabalho futuro, descrita no ponto 3.6.7, é a utilização de dados públicos guardados na base de dados MongoDB por alguma ferramenta de apoio à decisão, como por exemplo o Power BI.

Para finalizar, este trabalho contribuiu para o conhecimento sobre o cadastro nacional e a administração de propriedades com uma forte componente geográfica. Estes avanços contribuem em áreas com as maiores carências nos atuais sistemas nacionais. Este conhecimento detalhado e funcional da propriedade é crítica nos atuais modelos de inovação e governança territorial.

6. REFERÊNCIAS

- Cruz, E. F., & da Cruz, A. R. (2020). Design science research for is/it projects: Focus on digital transformation. *Iberian Conference on Information Systems and Technologies CISTI*.
- da Rocha, H. (2019). *Learn Chart.js*. Packt Publishing.
- de Beires, R. S., Amaral, J. G., & Ribeiro, P. (2013). *O Cadastro e a Propriedade Rústica em Portugal*. Fundação Francisco Manuel dos Santos.
- Direção-Geral do Território. (30 de Julho de 2021). *O snig e a implementação da diretiva INSPIRE*. Obtido de SNIG: <https://snig.dgterritorio.gov.pt/saber-mais/inspire/diretiva-e-disposicoes-execucao>
- European Commission. (30 de Julho de 2021). *Inspire consolidated uml model*. Obtido de Inspire Knowledge Base: <https://inspire.ec.europa.eu/data-model/approved/r4618-ir/html/index.htm?goto=2:1:3:1:7204>
- G. do Sec. de Estado dos Assuntos Parlamentares. (2018). *Relatório Sistema Informação Cadastral Simplificado*. República Portuguesa.
- Hanchett, E., & Listwon, B. (2018). *Vue.js en Action*. Manning Publications.
- Hows, D., Membrey, P., & Plugge, E. (2019). *Introdução ao MongoDB*. São Paulo: Apress Novatec.
- I. T. W. Group. (2014). *Infrastructure for spatial information in europe, data specification on cadastral parcels*. Obtido de INSPIRE: http://inspire.jrc.ec.europa.eu/documents/Data_Specifications/INSPIRE_DataSpecification_AD_v3.0.pdf
- ISO. (2012). *Iso 19152:2012*. Obtido de iso.org: <https://www.iso.org/standard/51206.html>
- Juszczak, J. (30 de Julho de 2021). *Getting Started*. Obtido de Vue-chartjs: <https://vue-chartjs.org/guide/>
- knexjs.org. (30 de Julho de 2021). *Knex.js*. Obtido de Knex.js: <https://knexjs.org/>
- Koutifaris, A. (2018). *Vuex Quick Start Guide*. Packt Publishing.

Leider, J., & Leider, H. (30 de Julho de 2021). *What is Vuetify*. Obtido de Vuetify:
<https://vuetifyjs.com/en/introduction/why-vuetify/#getting-started>

Lemmen, C., van Oosterom, P., & Bennett, R. (2015). The Land Administration domain Model. *Land Use Policy*.

Matt Zabriskie. (30 de Julho de 2021). *Introdução | Axios Docs*. Obtido de Axios:
<https://axios-http.com/ptbr/docs/intro>

MDN Web Docs. (30 de Julho de 2021). *Introdução Express/Node*. Obtido de Mozilla: https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction

OpenJS Foundation. (30 de Julho de 2021). *Node.js Readme*. Obtido de Github:
<https://github.com/nodejs/node/blob/master/README.md>

OpenJS Foundation. (30 de Julho de 2021). *Sobre Node.js*. Obtido de Node.js:
<https://nodejs.org/pt-br/about/>

Openlayers.org. (2021). *Openlayers*. Obtido de Openlayers:
<https://openlayers.org/>

PostGIS Project Steering Committee. (30 de Julho de 2021). *Spatial and Geographic objects for PostgreSQL*. Obtido de PostGIS: <https://postgis.net/>

PostGIS Project Steering Committee. (1 de Julho de 2021). *ST_PointOnSurface*. Obtido de PostGIS: https://postgis.net/docs/ST_PointOnSurface.html

Silva, Â. (2012). O Cadastro Português - a evolução e o momento actual de uma das peças fundamentais na gestão do território nacional. *Faculdade de Letras - Universidade de Coimbra*, p. 136.

Subramanian, H., & Raj, P. (2019). *Hands-On RESTful API Design Patterns and Best Practices*. Packt Publishing.

The PostgreSQL Global Development Group. (30 de Julho de 2021). *What is PostgreSQL*. Obtido de PostgreSQL: <https://www.postgresql.org/about/>

Vershinin, V. (30 de Julho de 2021). *Overview*. Obtido de VueLayers:
<https://vuelayers.github.io/#/?id=vuelayers>

Vuejs.org. (30 de Julho de 2021). *O que é Vuex*. Obtido de Vuex:
<https://vuex.vuejs.org/>

Worsley, J. C., & Drake, J. D. (2002). *Practical PostgreSQL*. United States of America: O'Reilly & Associates Inc.

You, E. (30 de Julho de 2021). *What is Vue.js*. Obtido de Vue.js:
<https://vuejs.org/v2/guide/>