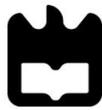




Universidade de Aveiro
2022

Ana Sofia Medeiros de Castro Moniz Fernandes **SIRP - Sistema de Informação do Registo de Propriedade**



Universidade de Aveiro
2022

**Ana Sofia Medeiros de
Castro Moniz
Fernandes**

**SIRP - Sistema de Informação do Registo de
Propriedade**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática, realizada sob a orientação científica do Professor Doutor Joaquim Manuel Henriques de Sousa Pinto, Professor Auxiliar do Departamento de Eletrónica, Telecomunicações e Telemática da Universidade de Aveiro

Dedico este trabalho aos meus dois melhores amigos, as estrelas mais brilhantes do céu.

o júri

presidente

Professora Doutora Maria Beatriz Alves de Sousa Santos
Professora Associada C/ Agregação da Universidade de Aveiro

vogais

Professor Doutor Fernando Joaquim Lopes Moreira
Professor Catedrático da Universidade Portucalense Infante D. Henrique

Professor Doutor Joaquim Manuel Henriques de Sousa Pinto
Professor Auxiliar da Universidade de Aveiro

agradecimentos

Nenhum caminho se percorre sozinho. Tenho tanto a agradecer que não cabe nestas palavras que vão resumir, em pouco, o muito que sinto.

Agradeço ao meu orientador. Todas as oportunidades que me proporcionou ao longo destes anos e por me ensinar que tudo se faz “por partes”. Obrigada.

Agradeço aos meus pais. Os meus pais, que me deram todas as ferramentas (e mais) para chegar aqui. Porque “não é como se cai, é como se levanta”. Obrigada.

Agradeço ao meu namorado, que sempre me fez ver que o caminho era menos difícil do que realmente parecia e que sempre me apoiou. Obrigada.

Agradeço ao meu amigo Alegria, que me ajudou desde o dia em que me tornei aluna do DETI e que leu esta tese na sua íntegra. Obrigada.

Por fim, agradeço a todos os meus colegas de mestrado que caminharam comigo. Sofremos juntos; rimos juntos. Obrigada.

“It all seems impossible until it’s done.”

palavras-chave

ASP.NET, OCR, React Native, Registo de Propriedade, São Tomé e Príncipe, Sistema de Informação.

resumo

Nos dias que correm, cada vez mais nos encontramos rodeados de tecnologias emergentes, que não param de evoluir. Esta evolução tecnológica traz consigo outras evoluções consequentes, como é o caso do registo de propriedade. Inúmeros países já realizaram a reforma digital, neste âmbito, o que trouxe diversas vantagens – não só é poupado tempo, em pesquisas que eram feitas em livros de papel (que acabam por ficar desgastados e mesmo desintegrados, ao longo do tempo), como são evitados muitos erros que acabam por ser cumulativos, quando registados em papel.

Em São Tomé e Príncipe, este registo ainda é feito em papel, em livros que chegam a contar com mais de dois séculos de existência. Assim, é necessária uma intervenção urgente nesta área, por forma a que o registo de propriedade comece a ser feito de forma eficiente, eficaz e segura. Esta mudança também trará inúmeros benefícios para a economia do país.

Com o Sistema de Informação do Registo de Propriedade é possível reunir, num único ambiente, todas as operações relacionadas com o registo de propriedade, tanto predial como automóvel, e garantir que todas ficam registadas em segurança.

keywords

ASP.NET, Information System, OCR, Property Registration, React Native, São Tomé and Príncipe.

abstract

Nowadays, we find ourselves increasingly surrounded by emerging technologies that do not stop evolving. This technological evolution brings with it other consequential evolutions, as happens with property registration. Many countries have already carried out the digital reform in this area, which has brought several advantages - not only is time saved in searches that used to be made in paper books (which end up being worn out and even disintegrating over time), but many errors that end up being cumulative, when registered on paper, are also avoided.

In São Tomé and Príncipe, this registration is still done on paper, in books that are up to two centuries old. Thus, urgent intervention is needed in this area, so that property registration can be done efficiently, effectively, and securely. This change will also bring countless benefits to the country's economy.

With the Sistema de Informação do Registo de Propriedade it is possible to bring together, in a single environment, all operations related to property registration, both land and automobile related, and ensure that all are securely recorded.

Conteúdo

Índice de Figuras	iii
Índice de Tabelas.....	vi
Lista de Acrónimos.....	vii
1 Introdução.....	1
1.1 Contexto e Motivação.....	1
1.2 Objetivos.....	3
1.3 Metodologias de Desenvolvimento de Software	5
1.4 Tecnologias Utilizadas.....	5
2 Estado Da Arte.....	7
2.1 Países que realizaram a reforma digital.....	9
2.2 Soluções existentes.....	10
2.2.1 Software para registo de terrenos em Cabo Verde.....	11
2.2.2 BUPi – Balcão Único do Prédio	12
2.2.3 Singapore Land Authority	13
2.2.4 NSW Land Registry Services.....	15
2.2.5 DORIS	15
2.2.6 Automóvel On-line	16
2.2.7 Registo de terrenos na Nigéria.....	17
2.2.8 eVRS.....	18
2.2.9 DUApp	19
2.2.10 Qual veículo é.....	20
2.3 Comparação Entre Soluções – Registo Automóvel.....	21
3 Tecnologias para Desenvolvimento do Software.....	23
3.1 Base De Dados	23
3.1.1 Base de Dados Relacional.....	23
3.1.2 Base de Dados Não Relacional	24
3.2 Tecnologias para o Desenvolvimento de Aplicações Web.....	26
3.2.1 Framework .Net.....	26
3.2.2 .NET Core	27
3.2.3 React.js.....	28
3.3 Tecnologias para o Desenvolvimento de Aplicações Móveis	28
3.3.1 React Native.....	28
3.3.2 Flutter	29
3.3.3 Android Studio	30
4 Levantamento de Requisitos.....	31
4.1 Contexto	31
4.2 Requisitos Funcionais.....	32
4.3 Requisitos Não Funcionais.....	34

4.4	<i>Requisitos Relativos ao Registo Automóvel</i>	35
4.5	<i>Perfis de Utilização</i>	37
4.6	<i>Personas</i>	38
4.7	<i>User Stories</i>	38
4.7.1	Aplicação Web.....	38
4.7.2	Aplicação Móvel.....	41
5	Implementação da Solução	42
5.1	<i>Escolha das Tecnologias</i>	42
5.2	<i>Arquitetura do Sistema</i>	42
5.3	<i>Estrutura da Base de Dados</i>	43
5.4	<i>Aplicação Web</i>	44
5.4.1	Registo predial.....	44
5.4.2	Registo automóvel.....	45
5.5	<i>Aplicação Móvel – Sirp Auto</i>	50
5.5.1	Propósito da Aplicação.....	50
5.5.2	Especificações Iniciais.....	51
5.5.3	Responsividade da Aplicação.....	52
5.5.4	Acesso à Aplicação – token de Autenticação.....	52
5.5.5	Reconhecimento Ótico de Caracteres.....	53
5.5.6	Leitura de QRCode.....	59
5.5.7	Modo Escuro e Modo Claro.....	61
5.5.8	Frequently Asked questions.....	63
5.5.9	Geração dos Instaláveis.....	64
6	Teste e Qualidade de Software	65
6.1	<i>Contexto</i>	65
6.2	<i>Testes de Integração</i>	66
6.3	<i>Testes Funcionais</i>	70
6.4	<i>Controlo de Versões</i>	74
7	Conclusões	76
7.1	<i>Trabalho Desenvolvido</i>	76
7.2	<i>Dificuldades Sentidas</i>	77
7.3	<i>Trabalho Futuro</i>	77
8	Referências	79
	Anexo 1 – Personas	84
	Anexo 2 – Diagramas de Sequência (Registo Automóvel)	85
	Anexo 3 – Estrutura da Base de Dados	87
	Anexo 4 – Registo Predial (Aplicação Web)	90
	Anexo 5 – Registo Automóvel (Aplicação Web)	92
	Anexo 6 – Relatório de Cobertura de Testes de Integração	99

ÍNDICE DE FIGURAS

Figura 1 - Livros referentes ao registo de propriedade, em São Tomé e Príncipe	2
Figura 2 - Livro relativo a registo de propriedade, em estado de degradação	3
Figura 3 - Digitalizador a fazer controlo de qualidade dos livros de registo de propriedade já digitalizados	4
Figura 4 - Tempo médio para o registo de uma propriedade, com e sem reforma informática	8
Figura 5 - Estágios necessários para digitalizar o registo de propriedades	8
Figura 6 - Tempo médio necessário para o registo de uma propriedade, por região	10
Figura 7 - Mudanças, em Cabo Verde, no sistema de registo de propriedades, antes e depois da reforma digital	11
Figura 8 - Página inicial do site BUPi.....	12
Figura 9 - Página inicial do site Singapore Land Authority	13
Figura 10 - Serviços online, oferecidos pelo Singapore Land Authority.....	14
Figura 11 - Portal online do NSW Land Registry Services	15
Figura 12 - Página inicial do site DORIS	16
Figura 13 - Página inicial do site Automóvel On-line.....	17
Figura 14 - Página inicial do site eVRS.....	19
Figura 15 - Página inicial da aplicação DUApp	20
Figura 16 - Página principal da aplicação "Qual Veículo É" e possível escolha de país	20
Figura 17 - Diagrama referente ao padrão MVC	27
Figura 18 - Requisitos para a criação do registo automóvel	36
Figura 19 - Diagrama de casos de uso (aplicação móvel)	41
Figura 20 - Arquitetura do sistema	43
Figura 21 - Página para consulta de máscaras de catalogação	46
Figura 22 - Fluxograma do processo de edição de um registo	47
Figura 23 - Fluxograma relativo à criação de um averbamento	49
Figura 24 - JSON com informação emulada, relativa a uma matrícula	51
Figura 25 - Página inicial da aplicação (Android e iOS, respetivamente) ...	53
Figura 26 - Permissões necessárias para ser possível aceder à câmara, em Android.....	54
Figura 27 - Permissões necessárias para ser possível aceder à câmara, em iOS	54

Figura 28 - Funções para leitura de imagem, através do uso da biblioteca "React Native Image Crop Picker"	55
Figura 29 - Uso da biblioteca "React Native MLKit OCR", dentro da função "getText"	56
Figura 30 - Opções de pesquisa por imagem de matrícula (Android e iOS, respetivamente)	56
Figura 31 - Escolha de imagem e recorte da mesma (opcional).....	57
Figura 32 - Informação relativa à matrícula pesquisada (Android e iOS, respetivamente)	57
Figura 33 - Exemplo de duas matrículas, em São Tomé e Príncipe	58
Figura 34 - Mensagem de erro - matrícula inválida (Android e iOS, respetivamente)	59
Figura 35 - Permissões necessárias para ser possível aceder ao armazenamento externo, em Android.....	59
Figura 36 - Leitor de QRCode, através da câmara do dispositivo (Android e iOS, respetivamente)	60
Figura 37 - Tendência de pesquisas por "dark mode", no Youtube	61
Figura 38 - Tendência de pesquisas por "dark mode", no Google.....	61
Figura 39 - Aviso inicial, relativo ao modo escuro ativo no dispositivo (Android e iOS, respetivamente)	62
Figura 40 - Página de pesquisa em modo escuro (Android e iOS, respetivamente)	63
Figura 41 - Página de FAQ (Android e iOS, respetivamente)	64
Figura 42 - Testes de integração realizados sobre os controladores.....	67
Figura 43 - Exemplo de teste onde é usado o Mock, para emular um utilizador, diretórios e ficheiros (parte 1)	68
Figura 44 - Exemplo de teste onde é usado o Mock, para emular um utilizador, diretórios e ficheiros (parte 2)	68
Figura 45 - Método Initialize(), presente nos testes de integração ao controlador Images	69
Figura 46 - Método Cleanup(), presente nos testes de integração ao controlador Images	70
Figura 47 - Testes funcionais realizados sobre a solução	70
Figura 48 - Método StartBrowser.....	73
Figura 49 - Método CloseBrowser	73
Figura 50 - Teste funcional para verificação da existência da opção de inscrição, nos registos do tipo "descrição"	73
Figura 51 - Repositório "SIRP", no GitLab	74
Figura 52 - Ficheiro read.me presente no repositório SIRP	75
Figura 53 - Pipeline criada para automatizar o processo de publicação do relatório de testes	75

Figura 54 - Diagrama de sequência relativo ao registo inicial e emissão de DUA	85
Figura 55 - Diagrama de sequência relativo à alteração das características físicas de um veículo	86
Figura 56 - Diagrama de sequência relativo à transmissão de um veículo... 86	
Figura 57 - Diagrama da base de dados do sistema.....	89
Figura 58 - Página para consulta de imagem dos livros	90
Figura 59 - Página para catalogação de uma imagem	90
Figura 60 - Página para consulta dos livros de registo	91
Figura 61 - Página de detalhes de uma máscara de catalogação	92
Figura 62 - Página relativa aos campos de uma máscara de catalogação....	92
Figura 63 - Página para criação de um registo automóvel inicial	93
Figura 64 - Página para consulta do livro de registos	93
Figura 65 - Página de edição de um registo automóvel inicial	94
Figura 66 - Página de edição de um averbamento (registo "singular").....	94
Figura 67 - Página para escolha do tipo de inscrição a realizar	95
Figura 68 - Página de visualização de detalhes de um registo.....	95
Figura 69 - Página para escolha do tipo de averbamento a realizar.....	96
Figura 70 - Página para validação de um registo	96
Figura 71 - Registo 255 validado	97
Figura 72 - Registo 255, após edição	97
Figura 73 - Página para eliminação de um registo	98
Figura 74 - Relatório gerado através do Fine Code Coverage, relativo à cobertura de código (parte 1)	99
Figura 75 - Relatório gerado através do Fine Code Coverage, relativo à cobertura de código (parte 2)	100

ÍNDICE DE TABELAS

Tabela 1 - Comparação entre soluções de registo automóvel.....	21
Tabela 2 - Requisitos funcionais (aplicação web) e respetiva descrição.....	33
Tabela 3 - Requisitos funcionais (aplicação móvel) e respetiva descrição....	34
Tabela 4 - Requisitos não funcionais (aplicação web e móvel) e respetiva descrição	35
Tabela 5 - Perfis de utilização do sistema	37
Tabela 6 - Testes funcionais realizados ao sistema	71
Tabela 7 - Personas existentes no sistema.....	84
Tabela 8 - Descrição das tabelas criadas na base de dados	87

LISTA DE ACRÓNIMOS

- SIRP** - Sistema de Informação do Registo de Propriedade
- DGRN** - Direção Geral dos Registos e Notariado
- API** – *Application Programming Interface*
- SDLC** – *Software Development Life Cycle*
- SGBD** – Sistema Geral de Base de Dados
- ACID** – Atomicidade, Consistência, Isolamento e Durabilidade
- JSON** – *JavaScript Object Notation*
- CLR** – *Common Language Runtime*
- FCL** – *Framework Class Library*
- MVC** – *Model View Controller*
- ITT** – Instituto dos Transportes Terrestres
- US** – *User Story*
- SDK** – *Software Development Kit*
- OCR** – *Optical Character Recognition*
- FAQ** – *Frequently Asked Questions*

1 INTRODUÇÃO

Neste capítulo serão apresentados o contexto e motivação que levaram à criação do tema desta dissertação, bem como os objetivos da mesma (incluindo as tecnologias a serem usadas na sua elaboração).

1.1 CONTEXTO E MOTIVAÇÃO

Em países desenvolvidos, como Portugal, a maioria das pessoas têm tendência a dar como garantidas certas coisas que noutras partes do globo, menos desenvolvidas, ainda são uma miragem do futuro. O registo de propriedade é uma das realidades indiscutíveis, para nós, mas que ainda não são universais. Apesar de Portugal se mostrar como um dos países que mais tem evoluído no registo de propriedade [1], ainda apresenta 153 municípios sem registo predial [2].

Mesmo sendo um direito de todos, conforme consagrado na carta das Nações Unidas[3], muitos países ainda não são capazes de reunir as condições necessárias para que os seus cidadãos possam proteger os seus direitos enquanto proprietários de bens. Note-se que no registo de propriedades inclui-se o registo de prédios (habitações e terrenos), veículos, embarcações e aeronaves. Nos países em que essa prática já está enraizada, a maioria deles continuam a ter um sistema de registo de propriedades exclusivamente feito em papel. Atente-se que o registo de propriedades ajuda a suportar o investimento, produtividade e crescimento de um país. Mesmo para os próprios governos, é essencial ter informação e registos sempre atualizados, por forma a poderem avaliar e cobrar de forma correta as taxas associadas a uma propriedade [4].

Como tentativa de melhorar a resposta aos problemas associados com os registos de propriedade em papel, os países têm apostado em processos de digitalização e informatização dos seus registos de propriedade. Esta alteração é também potenciada pela quarta revolução industrial [5], visto que cada vez mais nos encontramos rodeados de tecnologias que não param de evoluir e que podem, na maioria dos casos, ser uma mais-valia.

Um registo realizado em papel, além de poder demorar imenso tempo a ser efetivado (pode, mesmo, em alguns casos demorar-se meses para concluir um registo), também gera várias falhas, falhas estas que podem ser cumulativas e às quais rapidamente se perde o rasto. Um registo em papel, visto derivar de muitas pesquisas realizadas também em papel (e em livros que podem ter já centenas de

anos), pode acabar por não assegurar ao proprietário as garantias a que tem direito, quando faz o registo do seu bem.

Desta forma, têm sido reunidos esforços, direcionados principalmente para a criação de um guião de procedimentos para uma reforma digital do setor do registo de propriedade, existindo organizações que ajudam os países menos desenvolvidos a dar este passo. Com um sistema de registo digital, são evitados muitos erros, é poupado tempo, e é assegurada a segurança de registos, ao proprietário, visto que passam a ser registados automaticamente numa base de dados. Além disso, conseguem reunir-se várias funcionalidades num só sistema. As vantagens enumeradas são muitas, mas não podemos também esquecer os problemas associados com o acesso indevido aos registos e a possibilidade de deturpação/adulteração dos registos, muito mais difíceis de detetar que em livros em papel.

São Tomé e Príncipe, país pertencente à África Subsariana e que conta com cerca de 219 mil habitantes [6] (com uma densidade populacional de 228.3 pessoas por quilómetro quadrado de terreno [7]), ainda apresenta o seu registo de propriedade no formato de papel. Esses registos remontam ao período colonial, altura em que foi iniciado o registo de propriedades, que vem sendo mantido e atualizado até aos dias de hoje.



Figura 1 - Livros referentes ao registo de propriedade, em São Tomé e Príncipe



Figura 2 - Livro relativo a registo de propriedade, em estado de degradação

Todos os dias são catalogados, à mão, dezenas de registos. Estão registados, em papel, registos com mais de dois séculos. Como se pode observar a partir da Figura 1, são inúmeros os livros de registo de propriedade, que se foram acumulando ao longo do tempo. Além de serem muitos, mais de uma centena, estes livros, principalmente os de manuseamento mais comum, os com idade entre os cinco e os dez anos, acabam por ficar muito degradados. Como se pode observar na Figura 2, alguns já são impossíveis de serem processados e/ou lidos.

Note-se que quando, por exemplo, um terreno é passado, por herança de pai para filhos ou no caso de a propriedade ser dividida, é preciso recorrer ao registo anterior para se poder averbar a transferência de propriedade e, criar novos registos das parcelas entretanto criadas. Como é de imaginar, além de atrasar o processo de registo, este método é bastante propenso a erros, erros estes que podem vir a ser descobertos já numa fase bastante tardia do registo e que, nesse intervalo de tempo, podem vir a causar outras incorreções [8].

1.2 OBJETIVOS

O objetivo desta dissertação surgiu como resultado de uma proposta de bolsa de investigação, lançada pelo Professor Doutor Joaquim Sousa Pinto. É nesse seguimento que assenta o desenvolvimento da solução proposta, onde se procura construir um sistema integrado de registo de propriedade, o SIRP, de modo a deixar em definitivo os livros e papel e a passar a realizar as atividades apenas de forma digital. Como anteriormente mencionado, a tarefa de registo de propriedade, em São Tomé e Príncipe, ainda é realizada com recurso a registos de papel, o que dificulta o acesso à informação e potencia a ocorrência de inúmeras falhas no processo de registo.



Figura 3 - Digitalizador a fazer controlo de qualidade dos livros de registo de propriedade já digitalizados

O SIRP vai, então, permitir:

- Reunir, num único ambiente e em processos integrados, as funcionalidades necessárias ao registo de propriedade e transações associadas ao ciclo de vida da mesma;
- A existência de mecanismos que tolerem futuras interoperabilidades com sistemas de informação secundários (Direção dos Impostos, Direção dos Serviços Geográficos e Cadastrais e ainda outros serviços integrados na Direção Geral dos Registos e Notariado);
- A existência de canais adequados para a interação com os utentes dos serviços, disponibilizando assim formas de comunicação e acompanhamento de todos os processos por parte dos envolvidos;
- O acompanhamento e suporte ao lançamento de serviços nos canais suportados.

Desta forma, serão criados canais de ligação ao cidadão por meio de aplicações *web* e móveis, será feita a integração com o *site* da DGRN e também a criação de um conjunto de *API's/Web Services* para integração com os sistemas legados, mencionados anteriormente, garantindo-se assim a interoperabilidade com os mesmos.

Pretende-se assim, desenvolver os sistemas de *backoffice*, *frontoffice* e *API* de interface com outros sistemas de governação eletrónica, bem como uma aplicação móvel, de nome SIRP Auto, destinada ao registo automóvel.

1.3 METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

No início de qualquer projeto de *software*, as equipas e organizações precisam primeiro de decidir qual a metodologia a utilizar. Para o desenvolvimento do SIRP, optou-se pela metodologia ágil.

Os projetos de *software* seguem uma metodologia de processos claramente definidos ou ciclo de vida de desenvolvimento de *software* (SDLC) para garantir que o produto final seja de alta qualidade. Um SDLC identifica as fases e o fluxo estruturado de uma fase para outra. Normalmente, existem seis a sete fases. [9][10]

A metodologia ágil é um tipo de abordagem incremental para o desenvolvimento de *software* com base em princípios que se concentram mais nas pessoas, nos resultados, na colaboração e nas respostas flexíveis à mudança. Em vez de planear à partida todo o projeto, o processo de desenvolvimento é dividido em pequenos incrementos concluídos em iterações ou prazos curtos. Cada iteração inclui todas as fases do SDLC, de modo que um produto funcional seja entregue no final. Após várias iterações, uma nova versão ou atualização do produto é disponibilizada. Desta forma, ajuda a produzir e gerir projetos de alta qualidade. [9][10]

O desenvolvimento ágil é uma abordagem baseada na equipa que privilegia a criação rápida de uma aplicação funcional com foco na satisfação do cliente. Ele define períodos temporais chamados *sprints* com uma duração típica de duas semanas. No início de cada *sprint*, é elaborada uma lista de entregas que é priorizada com base no interesse do cliente. No final do *sprint*, a equipa de desenvolvimento e o cliente revêm e avaliam o trabalho e essas notas servem para *sprints* futuros. [9][10]

1.4 TECNOLOGIAS UTILIZADAS

Para construção da solução SIRP, foram escolhidas as seguintes tecnologias:

- *SQL Server*, para desenvolvimento da base de dados. É de notar que esta tecnologia já havia sido escolhida, no âmbito do projeto, antes do começo desta dissertação;
- *Framework ASP.NET*, para desenvolvimento da aplicação *web* e da API, através do uso da linguagem de programação *C#*. É de notar que

esta tecnologia já havia sido escolhida, no âmbito do projeto, antes do começo desta dissertação;

- *React Native CLI*, para desenvolvimento da aplicação móvel. Esta tecnologia foi escolhida já no âmbito desta dissertação.

2 ESTADO DA ARTE

No segundo capítulo, será feito um estudo acerca da evolução do registo de propriedade, em diferentes regiões, bem como o impacto desta evolução um pouco por todo o mundo. Seguidamente, é elaborada uma análise acerca de soluções já existentes, no contexto do registo de propriedade em formato digital.

Nos dias que correm, encontramos-nos cada vez mais rodeados de tecnologias que não param de evoluir e de se transformar. Estamos a vivenciar a quarta revolução industrial, que se iniciou em meados do século passado e que passa por uma forma de descrever a divisão cada vez mais estreita que existe entre os mundos físicos, digitais e biológicos [5]. Como a tecnologia muda a nossa forma de viver, também o processo de digitalização contribui para a transformação do mercado [8]. Esta transformação, contudo, não decorre em todo o mundo à mesma velocidade. Alguns países continuam a ter atrasos neste processo de transformação, mantendo-se ainda nos métodos mais tradicionais. Um dos exemplos onde esse processo de transformação está atrasado é o dos registos de propriedade que continuam a ser feitos em papel e espalhados por diversos livros de diferentes tipologias – o que pode levar a litígios, burlas e disputas sobre propriedades [8], além do tempo útil que pode ser perdido em toda esta forma de trabalhar.

O direito à propriedade está consagrado na Declaração Universal dos Direitos do Homem [3] que, no seu Artigo 17.º refere que 1. Toda a pessoa, individual ou coletivamente, tem direito à propriedade; 2. Ninguém pode ser arbitrariamente privado da sua propriedade.

Na Figura 4 pode verificar-se que registar uma propriedade por via digital implica um tempo de registo bastante mais curto, tempo este que tem vindo a diminuir consideravelmente ao longo dos anos.

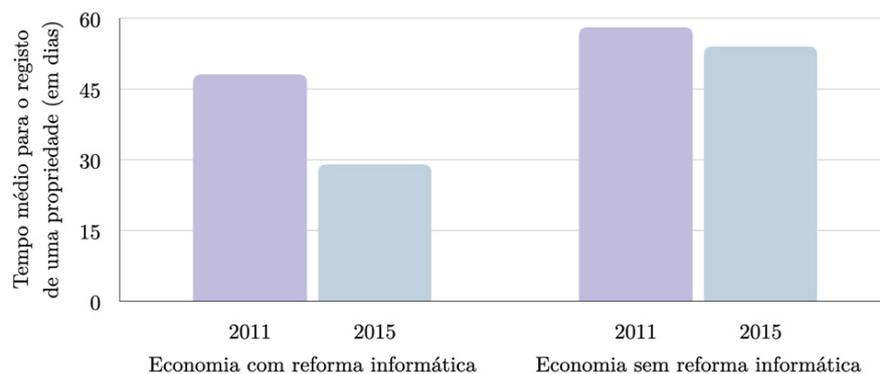


Figura 4 - Tempo médio para o registo de uma propriedade, com e sem reforma informática (adaptado de [9])

De acordo com os autores da publicação [9], as economias são beneficiadas em diferentes aspetos, aquando do processo de digitalização:

- Permite reduzir o espaço de armazenamento (deixa de ser físico), passando tudo a estar armazenado numa única base de dados;
- Melhora os serviços prestados;
- As transações passam a ser registadas de forma automática;
- Aumenta a segurança dos dados – ao contrário do que acontece com o papel, passam a existir *backups*;
- Diferentes pessoas podem aceder à mesma informação, ao mesmo tempo, em localizações diferentes.

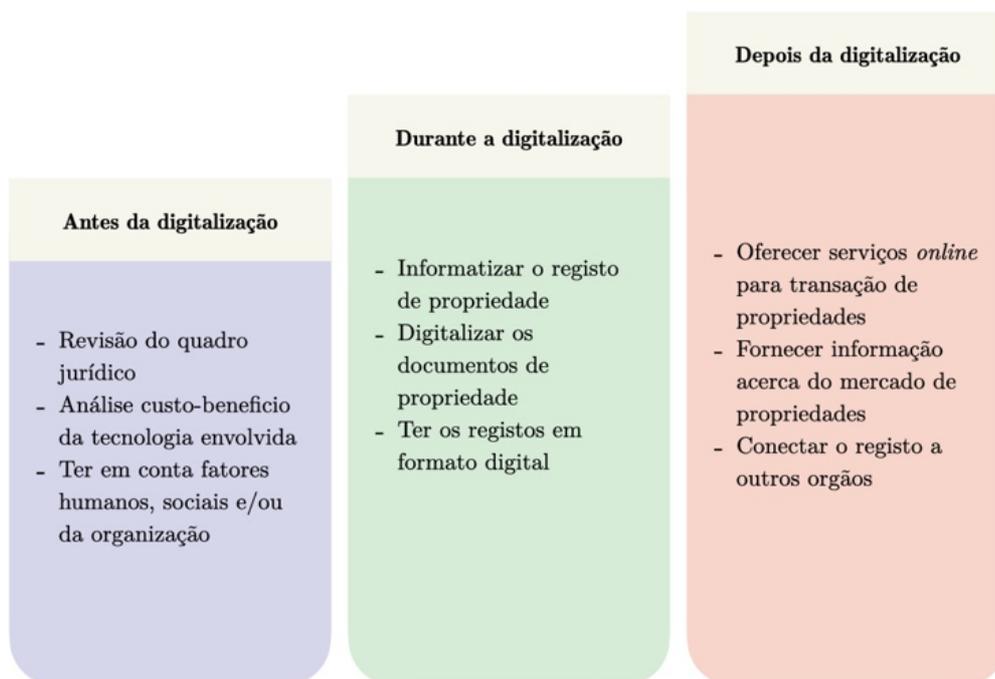


Figura 5 - Estágios necessários para digitalizar o registo de propriedades (adaptado de [9])

Como é possível de se observar na Figura 5, o processo de digitalização não tem que se restringir a uma única etapa, podendo ser um processo faseado. Estudos realizados em [10] sugerem que a aproximação gradual é a que mais eficaz se tem mostrado. Afirmam que países como a Nova Zelândia e a Noruega, que hoje em dia têm os sistemas de registo de propriedade mais eficientes do mundo, adotaram este método. Numa primeira fase, começaram por converter os registos em papel para o formato digital, informatizaram-nos e só depois introduziram o registo de propriedade eletrónico. Esta sequência não surge por acaso. Para que se possa fazer a transferência de uma propriedade de uma entidade para outra, é previamente necessário ter essa propriedade descrita e saber quem é o detentor atual da mesma. Assim, este processo tem de iniciar no primeiro registo e ir contruindo a sequência temporal dos seus donos até chegar ao dia de hoje [6].

Os autores deste mesmo estudo [10] reforçam também a importância que é ter um sistema eletrónico transparente, eficiente e acessível para o registo de propriedades, visto que propriedades não registadas não podem ser usadas como garantia para os bancos, limitando assim oportunidades financeiras.

Ainda assim, alguns países não têm possibilidades técnicas e/ou económicas para efetuar esta transição, pelo que organizações como o Banco Mundial¹ e a Organização dos Estados Americanos² têm estado envolvidos no processo [10].

2.1 PAÍSES QUE REALIZARAM A REFORMA DIGITAL

Dada a quantidade de vantagens já referidas, cada vez mais países optam por realizar uma reforma digital dos seus sistemas de registo de propriedade. As melhorias mais comuns, para quem escolhe fazer esta mudança, passam pela introdução de limites temporais e aceleração do processo de registo, o que melhora a eficiência administrativa e agiliza os processos [7].

Em 2010/11, cinco países escolheram o caminho da digitalização – Bélgica, República Checa, Eslovénia, Suazilândia e Vanuatu [10]. Já em 2011/12, cinco novos países se juntaram à lista – Bósnia e Herzegovina, Chipre, Itália, Maurícias e Polónia [7]. Isto levou a que se poupasse, em média, um mês no registo de propriedades. Além destes, também países como Portugal, Inglaterra, França, Holanda, Nova Zelândia, Samoa, Zâmbia, Angola, Gaza e Cabo Verde já apresentam um sistema digital para registo de propriedades [4][5].

Mesmo tendo o seu sistema já informatizado, os países não desistem de continuar a melhorar o processo. Em 2013 [9], o governo britânico deu

¹ <https://www.worldbank.org/en/home> (Acedido em 22 de julho de 2022)

² <http://www.oas.org/en/> (Acedido em 22 de julho de 2022)

quatrocentos dias para transformar vinte e cinco serviços (entre os quais o registo de propriedade), procurando torná-los mais simples, claros e rápidos. Isto levou a que, em 2013/14, a produtividade do registo predial aumentasse em 21% [9], comprovando, mais uma vez, que o processo de digitalização não se resume a uma única etapa, mas sim a um conjunto de ações faseadas e de melhoria contínua.

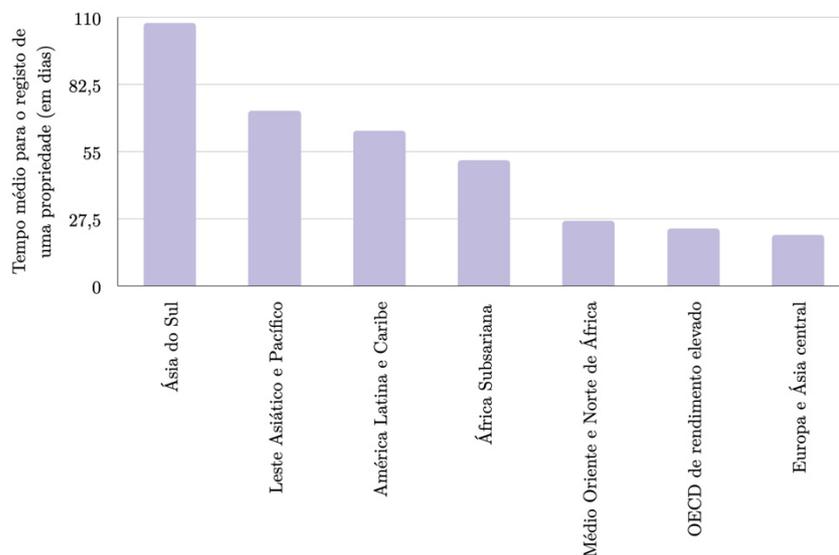


Figura 6 - Tempo médio necessário para o registo de uma propriedade, por região (adaptado de [13])

Como se pode observar na Figura 6, Ásia do Sul, o Leste Asiático e Pacífico, a América Latina e Caribe e a África Subsariana sobressaem pela negativa no registo de propriedades, sendo as quatro regiões onde o processo mais demora [13].

2.2 SOLUÇÕES EXISTENTES

De modo a melhor perceber o problema e definir uma linha de ação para a informatização de um registo de propriedade, serão analisadas algumas soluções existentes. Antes mesmo de iniciar a sua descrição, é possível constatar que todas as soluções apresentam vantagens relativamente ao processo de registo de propriedade não digitalizado. É de notar que também foram analisadas algumas aplicações móveis existentes, que lidam com informação relativa ao registo automóvel.

2.2.1 SOFTWARE PARA REGISTO DE TERRENOS EM CABO VERDE³

Mesmo pertencendo à África Subsariana, Cabo Verde mostra ser uma história de sucesso. Em novembro de 2012, foi lançado o *software* oficial para registo predial, resultado de um investimento de 17.3 milhões de dólares [14]. De acordo com a agência *Millennium Challenge Corporation*⁴ [14], o governo financiou a digitalização e catalogação dos livros de registos das ilhas de Santiago e do Sal, procurando diminuir o tempo para o registo e outras operações associadas a terrenos. Posteriormente, foi então construído um sistema de *software* para informatizar as transações em ambas as ilhas, sendo que a transição para o digital levou cerca de quatro meses.

Antes da digitalização	Depois da digitalização
<ul style="list-style-type: none">- Muito tempo despendido na procura de informação em livros- Registos escritos à mão- O período para entrega de registos era de cerca de 15 dias- Facilmente ocorriam erros na pesquisa de informação, que poderiam levar a erros no próprio registo	<ul style="list-style-type: none">- Todas as operações relacionadas com registo de propriedade estão agora inseridas no novo sistema de software- O período para entrega de registos passa para cerca de 3 dias- Já não ocorrem erros de registo derivados de pesquisas incorretas de informação em livros- Os clientes passam a gastar menos dinheiro em deslocações pois não sentem tanta necessidade de esclarecer informação presencialmente

Figura 7 - Mudanças, em Cabo Verde, no sistema de registo de propriedades, antes e depois da reforma digital (adaptado de [14])

Através da análise da Figura 7, é possível reter-se uma noção imediata das vantagens que a reforma digital acarretou. Além dos funcionários poderem agora aceder a toda a informação de forma bastante mais rápida, por estar

³ É de notar que, apesar de se encontrar esta informação em diversas páginas e artigos, não é possível aceder a um portal ou a um *software* para confirmar a veracidade do mesmo.

⁴ <https://www.mcc.gov> (Acedido em 22 de julho de 2022)

tudo centralizado num único sítio (o *software*), são ainda evitados erros que acabavam por acontecer no processo do registo, dado que toda a informação era pesquisada em livros, processo facilmente falível. Além disso, concluiu-se que uma reforma digital não trouxe vantagens unicamente aos funcionários ou à empresa em si. Também os clientes ficaram a ganhar porque acabaram por poupar dinheiro em deslocações, visto ser reduzida a necessidade de esclarecer informações presencialmente.

2.2.2 BUPi – BALCÃO ÚNICO DO PRÉDIO

Sistema português que surgiu em 2017 e que procurava conhecer o território português de forma simples e inovadora [15]. Propunha-se cumprir quatro objetivos [15]:

1. Ordenar o território;
2. Valorizar os recursos;
3. Identificar os proprietários;
4. Prevenir incêndios.



Figura 8 - Página inicial do site BUPi

Em suma, “O BUPi⁵ permite aos 174 municípios que não dispõem de cadastro predial disponibilizarem aos seus munícipes um serviço de georreferenciação simplificada, que encurta o tempo e os recursos necessários a este mapeamento.”. Assim, os direitos do proprietário ficam protegidos, visto

⁵ <https://bupi.gov.pt> (Acedido em 22 de julho de 2022)

que este regista o seu terreno na Conservatória, com reconhecimento dos limites geográficos. [15]

O cidadão pode também aceder a uma plataforma *online*, usando o seu cartão de cidadão ou chave móvel digital, permitindo através desta aceder às suas propriedades ou localizar uma nova, ficando assim dispensado de se dirigir a um balcão *BUPi*.

2.2.3 SINGAPORE LAND AUTHORITY

De acordo com a sua página *online* [16], o *Singapore Land Authority*⁶ “(...) é um conselho estatutário dependente do Ministério do Direito. Otimizam os recursos terrestres para o desenvolvimento económico e social de Singapura. Isto é importante uma vez que a terra é utilizada para uma vasta gama de atividades em Singapura.”. Desta forma, este conselho estatutário [16]:

- Retém a base de dados nacional, que contém a informação acerca dos terrenos de Singapura;
- Gere os terrenos do estado, através de vendas, arrendamentos, aquisições e alocações;
- Desenvolve e comercializa as propriedades do estado, otimizando o seu uso em benefício dos cidadãos de Singapura;
- É a autoridade nacional do registo de terrenos, sendo responsáveis pelo registo de transações das propriedades de Singapura;
- Gere e mantém o sistema nacional de levantamento de terras.

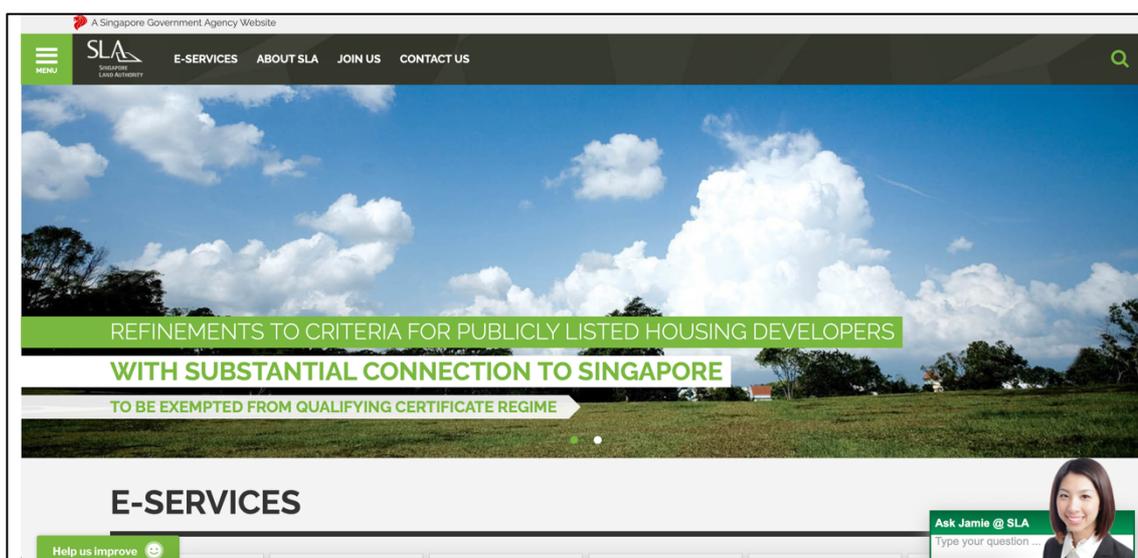


Figura 9 - Página inicial do site Singapore Land Authority

⁶ <https://www.sla.gov.sg> (Acedido em 22 de julho de 2022)

No entanto, o mais interessante e inovador neste sistema é o facto de o cidadão ter ao seu alcance um conjunto de operações *online* que facilitam todo o processo de registo/transação de propriedades. Um cidadão de Singapura conta com dez serviços online [17]:

1. Encontrar informação acerca de uma propriedade ou terreno;
2. Arrendar terreno ou propriedade do estado;
3. Comprar um terreno do estado;
4. Oficializar o título da propriedade;
5. Encontrar dados cartográficos através do *OneMap*;
6. Subscrever uma rede de referências (suporta posicionamento de alta precisão, em tempo real [18]);
7. Ver informação acerca da sua propriedade;
8. Requerer uma licença de ocupação temporária;
9. Submeter um inquérito cadastral;
10. Submeter um pedido à LDAU (*Land Dealings Approval Unit*).

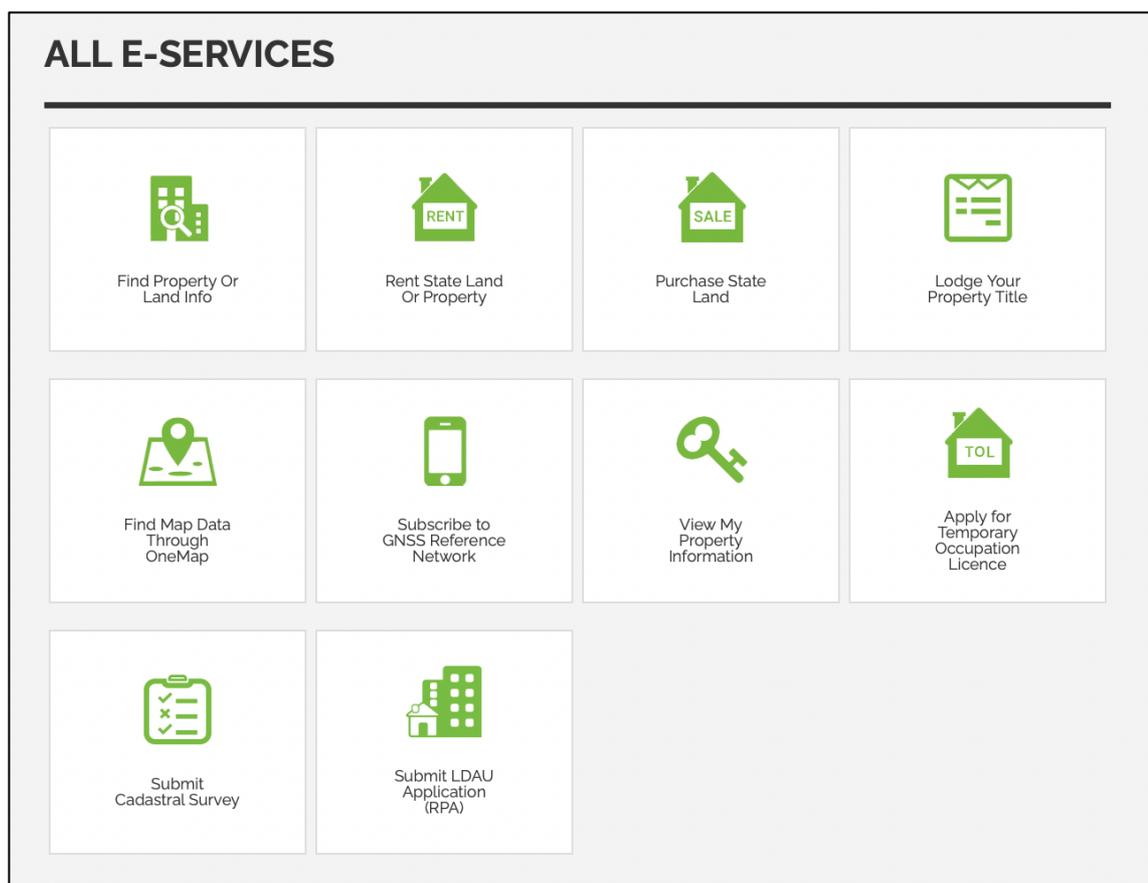


Figura 10 - Serviços online, oferecidos pelo Singapore Land Authority

2.2.4 NSW LAND REGISTRY SERVICES

Em Nova Gales do Sul, um estado do sudeste da Austrália, foi criado o *NSW Land Registry Services*⁷, que trata do registo dos títulos de propriedade dos terrenos, para o estado e para os cidadãos, sustentando assim um sistema seguro e eficiente [19]. De acordo com a sua página, “(...) o *NSW Land Registry Services* melhorou os resultados dos serviços para os clientes, investiu para aumentar a segurança dos registos e permitiu a inovação digital nos setores de propriedade e agrimensura.” [19].

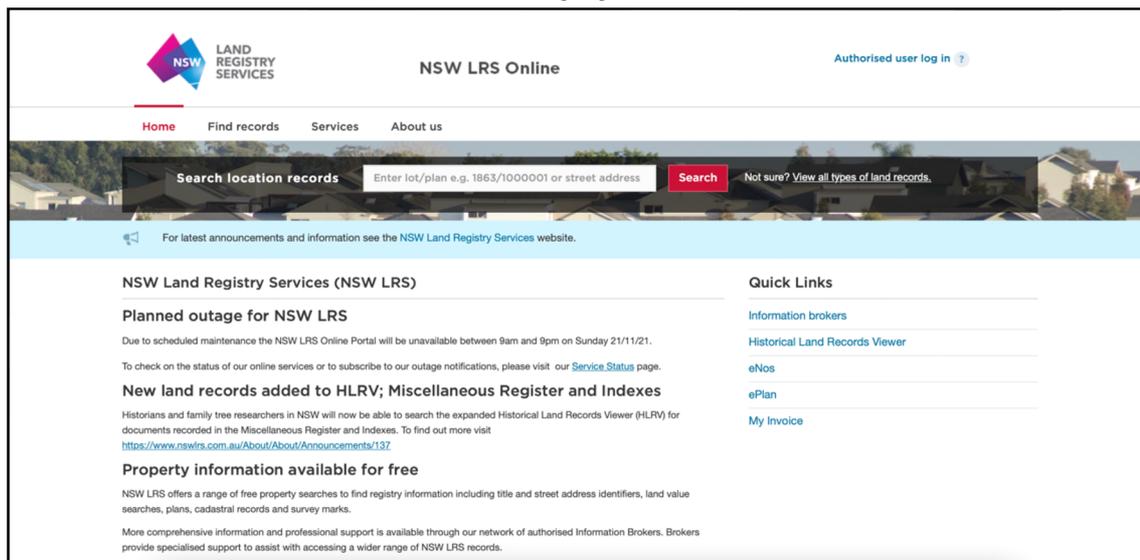


Figura 11 - Portal online do NSW Land Registry Services

Aqui, os cidadãos de Nova Gales do Sul contam com um portal *online* para que possam aceder a diferentes serviços relacionados com títulos de propriedade. À distância, podem ainda consultar *brokers* de informação autorizada para assistência profissional, procurar registos de títulos de propriedades e outros produtos, bem como aceder a um visualizador de históricos de registos de terrenos, onde é possível visualizarem-se alguns gráficos. Se preferirem, os cidadãos podem também dirigir-se a terminais públicos para serem ajudados nos serviços relativos a informação de títulos de propriedade. [20]

2.2.5 DORIS

De acordo com a Figura 6 anteriormente presente, pode observar-se que o sul da Ásia é a região que mais tempo leva para efetivar o registo de uma propriedade. Ainda assim, a Índia é um país desta região que se destaca depois de apostar em vários serviços digitais de registo de propriedade.

⁷ <https://www.nswlrs.com.au> (Acedido em 22 de julho de 2022)

O DORIS⁸, *Delhi Online Registration Information System*, é um sistema que, tal como o próprio nome indica, permite o registo de propriedades, através de um portal *online*. Este destaca-se pela quantidade de funcionalidades que consegue reunir. Nele, é possível [21]:

- Procurar documentos registados;
- Ver os documentos de escrituras registadas;
- Calcular o imposto de selo pago ao governo;
- Escrever documentos legais para escrituras;
- Fornecer serviços certificados como cópias certificadas, inspeções, etc.;
- Consultar a lista de propriedades que são proibidas;
- Consultar cotas a serem pagas;
- Registrar reclamações e queixas.



Figura 12 - Página inicial do site DORIS

2.2.6 AUTOMÓVEL ON-LINE

O conceito de propriedade não se limita apenas a terrenos. O registo de propriedade divide-se em registo predial - para terrenos, prédios, frações urbanas, registo de veículos, aeronaves e embarcações. Em Portugal, um dos países que mais rápido se tem mostrado aquando do registo de propriedade [7],

⁸ <https://doris.delhigovt.nic.in> (Acedido em 22 de julho de 2022)

os cidadãos podem contar com o *Automóvel On-line*⁹ para realizar o registo dos seus veículos. De acordo com a sua página, é possível [22]:

- Realizar o pedido *online* e consultar o respetivo estado, após realizado, do registo de transferência da propriedade do veículo;
- Realizar o pedido *online* e consultar o respetivo estado, após realizado, da certidão permanente do registo do veículo.

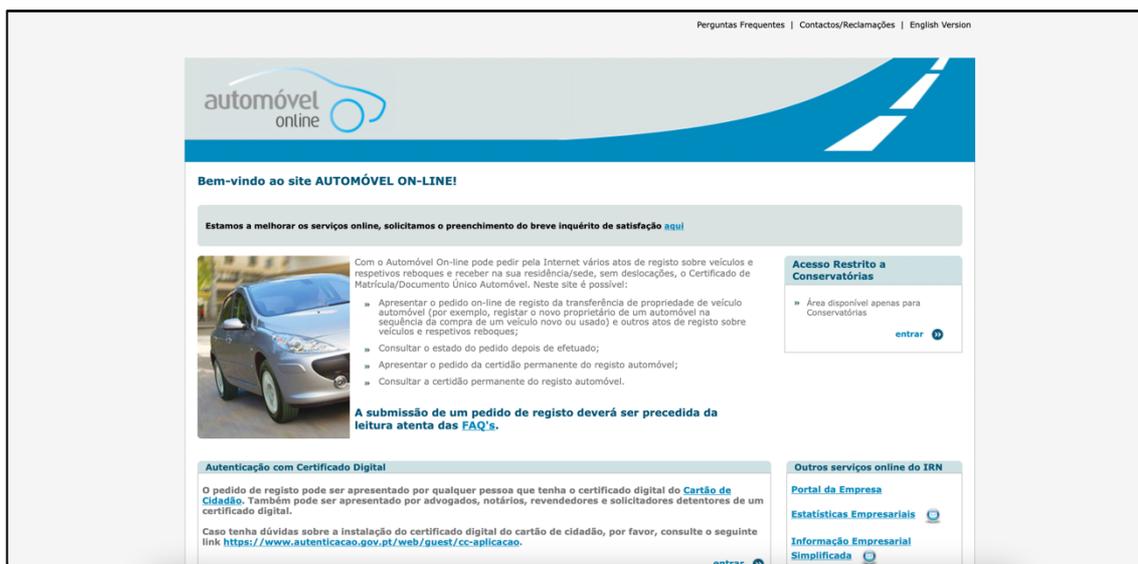


Figura 13 - Página inicial do site Automóvel On-line

É de notar que o indivíduo que realiza um pedido de registo *online* tem que, obrigatoriamente, apresentar o certificado digital do seu cartão de cidadão, sendo assim reforçada a segurança de todo o processo.

2.2.7 REGISTO DE TERRENOS NA NIGÉRIA¹⁰

Também a Nigéria, na África Subsariana, se tem demonstrado como uma história de sucesso. Tal como acontece nos restantes países, o registo de terrenos deve estar sempre atualizado com todas as transações que ocorreram até ao momento [6]. No estado de Lagos, em Fevereiro de 2005, começaram a digitalizar-se os documentos de propriedade dos terrenos – até 2015, haviam já sido digitalizadas e arquivadas 10.5 milhões de páginas [6]. Por conseguinte, a Nigéria acabou por fazer a transição do papel para o digital, criando um sistema eletrónico *online* de propriedade – em 2020, o país ficou classificado

⁹ <https://www.automovelonline.mj.pt/AutoOnlineProd/> (Acedido em 22 de julho de 2022)

¹⁰ É de notar que, apesar de se encontrar esta informação em diversas páginas e artigos, não é possível aceder a um portal ou a um *software* para confirmar a veracidade do mesmo.

na posição 131, de entre 190 países, quando se fala de facilidade em fazerem-se negócios, subindo assim quinze lugares no *ranking* desde o ano anterior [23].

De acordo com Franklin Okeke [23], a Nigéria beneficiou de diferentes formas, ao realizar a reforma digital:

- Poupança de tempo: como já mencionado anteriormente, existiam bastantes falhas e atrasos relacionados com registo de propriedades em papel. Com a reforma digital, isto deixou de acontecer, passando ainda a existir segurança de registos;
- Redução da corrupção: ao introduzir um sistema claro e transparente, a digitalização evitou casos como desaparecimentos misteriosos de registos ou casos em que uma propriedade tem mais que uma pessoa a reclamar a sua detenção;
- Segurança da informação e benefícios operacionais: como as transações são registadas automaticamente numa base de dados, sem grande interferência humana, o risco de erro acaba por ser mínimo. A eficiência operacional também é melhorada pois existe inclusão automática de informação, nas propriedades registadas, e permite também que diferentes departamentos do governo partilhem informação.

2.2.8 EVRS

Além do registo *online* de propriedade, a Nigéria conta também com um portal que permite aos cidadãos renovarem os documentos dos seus veículos de forma instantânea. O *eVRS*¹¹ (*Electronic Vehicle Registration System*) permite registos *online* e renovação dos mesmos, para todos os tipos de veículos, em toda a nação [24]. Assim, é possível [25]:

- Gerir os seus próprios veículos: verificar documentos expirados; modificar detalhes acerca do proprietário; adicionar veículos à sua própria conta; ver as transações realizadas e descarregar os documentos em *PDF*;
- Verificar o registo de veículos;
- Renovar documentos.

¹¹ <https://www.evrs.ng> (Acedido em 22 de julho de 2022)

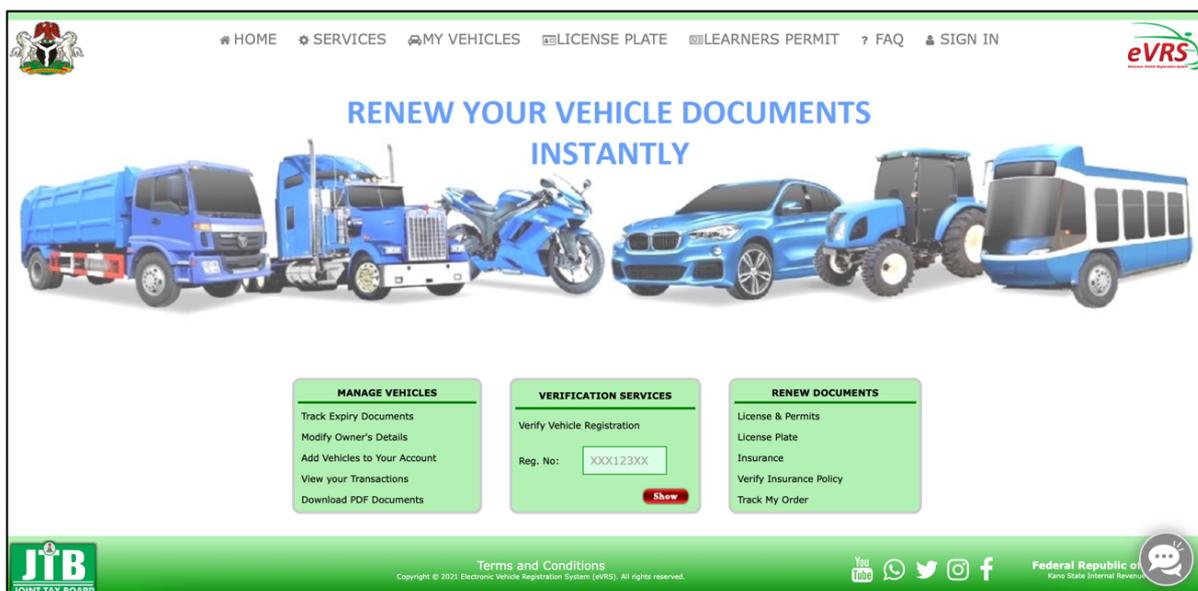


Figura 14 - Página inicial do site eVRS

2.2.9 DUAPP

A *DUApp* é uma aplicação móvel portuguesa, disponível tanto para *Android* como para *iOS*, que permite ler o *QRCode* presente no Documento Único Automóvel português e aceder a todas as informações, visíveis e não visíveis, que nele constam [26].

Apesar de esta aplicação se mostrar como uma inovação, no mercado de aplicações portuguesas, os utilizadores não se têm mostrado completamente satisfeitos – na *Google Play Store*¹² apresenta uma avaliação de 2.8, numa escala de 5, sendo que na *App Store*¹³ conta com 1.8, também numa escala de 5. Os comentários passam pela impossibilidade de abrir a aplicação, de realizar a leitura do *QRCode*, e até mesmo da falta de funcionalidades.

¹² https://play.google.com/store/apps/details?id=incm.com.leitor.dua&hl=pt_PT&gl=US (Acedido em 22 de julho de 2022)

¹³ <https://apps.apple.com/pt/app/duapp/id1474224031> (Acedido em 22 de julho de 2022)

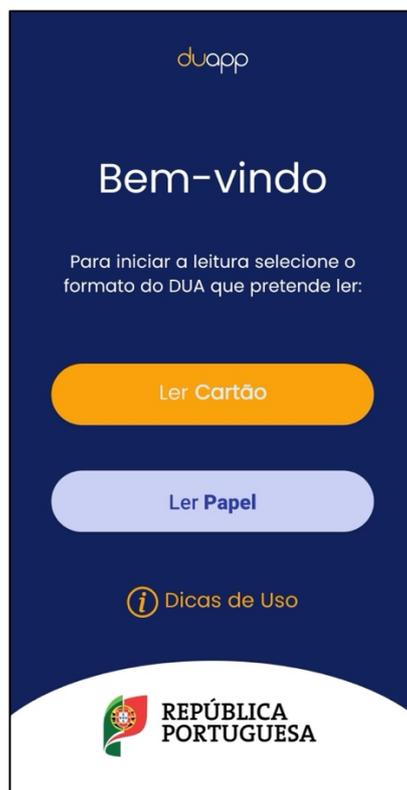


Figura 15 - Página inicial da aplicação DUApp

2.2.10 QUAL VEÍCULO É

A aplicação “Qual Veículo É” está disponível apenas para dispositivos *Android*, e permite pesquisar matrículas em todo o mundo ou num país específico, que pode ser selecionado numa lista.

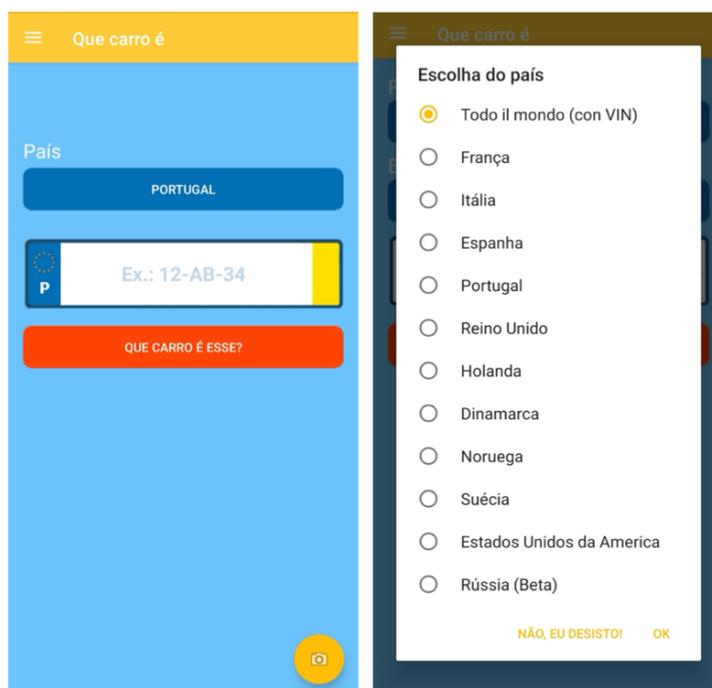


Figura 16 - Página principal da aplicação "Qual Veículo É" e possível escolha de país

Apesar de ser uma aplicação que conta com mais de um milhão de transferências (até à data), está avaliada com apenas 3.7 em 5 estrelas. A maior queixa dos utilizadores portugueses, presente nos comentários na *Google Play Store*¹⁴, é que, ao pesquisar-se por uma matrícula, não existe informação acerca da mesma. Ora, isto remove o propósito principal da aplicação – o de se conseguir obter informação a partir da pesquisa ou leitura (com a câmara) de uma matrícula.

2.3 COMPARAÇÃO ENTRE SOLUÇÕES – REGISTO AUTOMÓVEL

Dado que o foco principal desta dissertação será o registo automóvel, optou por se construir uma tabela comparativa entre as soluções que lidam com este tipo de registo, como mostra a Tabela 1. Desta forma, pode então concluir-se que o SIRP, tanto na sua versão *web* como móvel, é o que apresenta uma maior oferta de funcionalidades, dentro do registo automóvel.

No entanto, relembra-se que as soluções anteriormente estudadas não funcionam todas no mesmo país, sendo que nenhuma é utilizada em São Tomé e Príncipe.

Tabela 1 - Comparação entre soluções de registo automóvel

Funcionalidades /Solução	Automóvel On-line	eVRS	DUApp	Qual Veículo É	SIRP	SIRP Auto
Realizar pedido de transferência de propriedade do veículo	✓	✓	---	---	✓	---
Realizar pedido de certidão permanente do registo do veículo	✓	✓	---	---	✓	---
Renovar documentos	?	✓	---	---	✓	---
Registar novos veículos	✓	✓	---	---	✓	---
Leitura de QR Codes	---	---	✓	X	---	✓

¹⁴ https://play.google.com/store/apps/details?id=ro.gliapps.quellevoiture&hl=pt_PT&gl=US
(Acedido em 22 de julho de 2022)

Leitura de matrículas através do uso da câmara do dispositivo	---	---	X	✓	---	✓
Compatibilidade com os sistemas operativos <i>Android e iOS</i>	---	---	✓	X	---	✓

3 TECNOLOGIAS PARA DESENVOLVIMENTO DO SOFTWARE

Nesta secção, tendo em vista o desenvolvimento da solução, foram estudadas algumas tecnologias possíveis de serem usadas. É de notar que estas não param de aumentar e de evoluir, de dia para dia. Assim, por forma não estender demasiado o capítulo em questão, este estudo foi restringido aos diferentes tipos de base de dados, bem como tecnologias existentes para construção de aplicações web e móveis.

3.1 BASE DE DADOS

Uma base de dados é, tal como o próprio nome indica, um conjunto de informações/dados que se podem relacionar entre si. Estes dados podem ser de diferentes tipos, como por exemplo texto, numérico ou booleano.

Os Sistemas de Gestão de Base de Dados (SGBD) tomam, no desenvolvimento desta dissertação, um papel fundamental, visto serem sistemas de *software* que permitem gerir essa informação.

As bases de dados dividem-se em dois grandes grupos – relacionais e não relacionais.

3.1.1 BASE DE DADOS RELACIONAL

Uma base de dados relacional passa por uma coleção de elementos que apresentam relações, já pré-definidas. A informação está, assim, organizada em tabelas, que contêm linhas e colunas – cada coluna tem o seu tipo de dados, sendo que as linhas contêm os valores respetivos. Todos os dados de uma coluna devem ser do mesmo tipo, sendo que cada tabela deve ter um atributo ou combinação de atributos que identifica, de forma única, cada linha. [27]

Este tipo de base de dados apresenta quatro propriedades fundamentais, que asseguram a consistência da base de dados [28]:

1. Atomicidade: uma transação (unidade de lógica que acede à informação, a partir de operações de escrita e leitura) ocorre por completo ou nem sequer ocorre;
2. Consistência: uma transação resulta numa transformação correta da base de dados, mantendo a sua consistência;
3. Isolamento: várias transações podem ocorrer concorrentemente, sem interferirem umas com as outras e sem causarem inconsistência;
4. Durabilidade: assim que a transação termina, as atualizações feitas na base de dados são escritas e guardadas em disco, de forma persistente.

Estas quatro propriedades são, frequentemente referidas como ACID – Atomicidade, Consistência, Isolamento e Durabilidade.

3.1.1.1 SQL Server

SGBD relacional, desenvolvido pela *Microsoft*. Hoje em dia, está também disponível em *Linux* e através de *Docker*.

Apresenta várias vantagens – garante segurança da base de dados, é fácil de configurar e instalar, permite aceder à mesma base de dados através de diferentes dispositivos e apresenta funcionalidades para recuperação e restauração da informação [29]. A informação é acedida através de *queries*.

3.1.1.2 PostgreSQL

É também um SGBD relacional, mas orientada a objetos e bastante poderosa. Além de apresentar uma oferta muito vasta de funções, uma única instância de *PostgreSQL* pode conter mais de 4 biliões de base de dados – consegue suportar muito mais informação do que aquela que conseguimos sequer imaginar. É também bastante poderoso no que toca a integridade e concorrência no acesso aos dados. Suporta *queries* em *SQL* e em *JSON* (*JavaScript Object Notation*). [30]

3.1.2 BASE DE DADOS NÃO RELACIONAL

Este tipo de base de dados não usa o sistema de tabelas, com linhas e colunas – apresenta um modelo de dados flexível que pode ser de vários tipos, como por exemplo chave-valor ou *JSON* [31]. São também apelidadas de “No-

SQL”, que significa “*Non-Relational Database*”, e podem conter informação duplicada.

As bases de dados não relacionais apresentam várias vantagens [31]:

- Modelos de dados flexíveis: num modelo chave-valor, por exemplo, não é imposto nenhum tipo de dados, oferecendo muita flexibilidade para que a informação esteja de acordo com os requisitos da aplicação;
- Não apresentam esquemas, o que é o ideal para grandes quantidades de informação;
- Oferecem alta performance e escalabilidade.

Ainda assim, este tipo de base de dados não suporta transações, não havendo qualquer certeza de que a informação seja consistente. [32]

3.1.2.1 MongoDB

Base de dados *No-SQL* orientada a documentos, tradicionalmente usada para lidar com grandes volumes de informação. Estes documentos são constituídos por pares chave-valor, a unidade básica do *MongoDB*. Cada base de dados tem coleções que, por sua vez, têm documentos. Escala horizontalmente, repartindo a informação por várias instâncias de *MongoDB*, o que permite correr em diversos servidores. Assim, o seu propósito principal passa por permitir criar uma base de dados escalável, flexível e rápida. [33]

No entanto, mesmo apresentando alta performance, disponibilidade, velocidade e simplicidade, contém algumas limitações [34]:

- A única forma de fazer *join* (nas bases de dados relacionais, é uma operação que permite aglomerar informação de várias tabelas, com base nos seus campos) de documentos é adicionando código manualmente, o que pode ser uma tarefa penosa;
- Apesar do *indexing* oferecer altas velocidades, quando mal implementado, leva a que o *MongoDB* apresente velocidades de resposta mais baixos;
- Poderá exigir um alto uso de memória, uso este que pode derivar da falta da operação *join* que, conseqüentemente, cria a necessidade de duplicação de informação, causando assim redundância.

3.1.2.2 Apache Cassandra

É um SGBD distribuído e *open-source*, usado para gerir grandes quantidades de informação, que podem mesmo estar distribuídas por todo o mundo, apresentando alta disponibilidade. É altamente escalável, tolerante a falhas, consistente e orientado a colunas. Tem suporte para transações (permitindo ACID), sendo que as escritas na base de dados caracterizam-se por serem bastante rápidas. [35]

Mais uma vez, não sendo uma base de dados com sintaxe *SQL*, existem funções que não é capaz de suportar, como o *join*. Apesar das escritas serem rápidas, vários pedidos de leitura podem atrasar esta operação, tornando-se assim um processo demorado. Também há o risco de existir informação duplicada. [36]

As operações de *query*, para retorno de informação, são bastante limitadas, e a informação não é naturalmente ordenada – passa a ser uma decisão de *design* [37].

3.2 TECNOLOGIAS PARA O DESENVOLVIMENTO DE APLICAÇÕES WEB

Existe todo um leque de tecnologias para o desenvolvimento de aplicações *web*. Continuam a surgir novas, de ano para ano, e as existentes não param de evoluir, sendo que cada uma apresenta as suas vantagens.

3.2.1 FRAMEWORK .NET

Plataforma de desenvolvimento criada pela *Microsoft*, que permite criar e compilar aplicações destinadas ao sistema operativo *Windows*. Com esta *Framework*, os desenvolvedores de *software* não necessitam de escrever código para gerir a segurança, centralização ou transações [38]. Apresenta três principais componentes [38]:

1. *Common Language Runtime* (CLR): permite a execução de programas escritos na Framework, usando *C#*, *VB*, *Visual C++* ou outra linguagem suportada;
2. *Framework Class Library* (FCL): permite escrever programas *.NET* usando as linguagens suportadas;

3. Interfaces do programa e utilizador: fornece as ferramentas necessárias para a construção das aplicações – por exemplo, formulários *web* e serviços *web*.

ASP.NET

Permite estender a plataforma de desenvolvimento *.NET* com ferramentas e bibliotecas específicas para a construção de aplicações *web* [39].

Um aspeto importante, adicionado pelo *ASP.NET*, é a existência do padrão *Model View Controller* (MVC). Este vai permitir a criação de páginas *web* dinâmicas, a partir do desacoplamento da interface do utilizador (*View*), dados (*Model*) e a lógica da aplicação (*Controller*) – isto permite, assim, ter separação lógica dos componentes [40]. O design deste padrão é constituído por [41]:

1. *Model*: contém única e simplesmente os dados da aplicação, não contando com qualquer lógica;
2. *View*: apresenta os dados do *model* ao utilizador – sabe como aceder a essa informação, mas não sabe o que esta significa;
3. *Controller*: encontra-se entre o *model* e a *view* – ouve eventos desencadeados pela *view* ou outra fonte externa, toma a ação adequada e esta é mostrada na *view*.

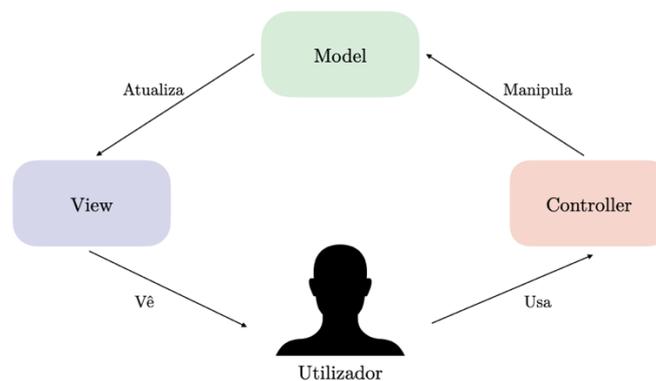


Figura 17 - Diagrama referente ao padrão MVC
(adaptado de [41])

3.2.2 .NET CORE

Lançada em 2016, *.NET Core* é uma plataforma para desenvolvimento de aplicações. Ao contrário da *Framework .NET*, que apenas permite o seu desenvolvimento em sistemas operativos *Windows*, agora é possível realizar este desenvolvimento também em *Linux* e *macOS*. Além disso, é *open-source*

e de alta performance. Permite ainda o desenvolvimento de aplicações móveis, algo até então impossível com a *Framework .NET*.

3.2.3 REACT.JS

Framework de desenvolvimento de aplicações *web*, desenvolvida pelo *Facebook*, baseada em *Javascript* (linguagem de programação) e *open-source*. Surgiu em 2013 e tem vindo a tornar-se cada vez mais popular no mundo da programação. É bastante fácil e intuitiva de se usar, principalmente para desenvolvedores que tenham alguma experiência na área das aplicações *web*. É frequentemente usado em casos em que a informação da aplicação está constantemente a atualizar.

Permite, assim [42][43]:

- Criar, facilmente, interfaces dinâmicas - ao contrário do *Javascript*, oferece mais funcionalidades usando-se menos código;
- Criar aplicações declarativas, modulares, rápidas e escaláveis usando *Javascript*;
- Melhorar a performance;
- Reutilizar componentes – uma vez criado um componente, pode ser reutilizado em toda a solução, não tendo que se refazer a sua lógica (mudam-se apenas os atributos, se necessário);
- Ter um fácil *debug* de erros e perceção dos mesmos, visto que o fluxo da informação é unidirecional.

3.3 TECNOLOGIAS PARA O DESENVOLVIMENTO DE APLICAÇÕES MÓVEIS

3.3.1 REACT NATIVE

Biblioteca *Javascript*, de acesso aberto, que permite desenvolver aplicações móveis, nativas, para *Android* e *iOS*. Uma aplicação móvel nativa é aquela que é instalada a partir da loja de aplicações presente no dispositivo, sendo desenhada especificamente para uma plataforma. Assim, a aplicação pode tirar partido de todas as funcionalidades do equipamento, como a câmara ou o sistema de notificações existente [44].

É baseado em *React*, partilhando assim das vantagens descritas acima, de *React.js* – é, também ele, baseado em componentes reutilizáveis. Apresentando

assim uma sintaxe bastante similar entre eles, torna-se muito fácil transformar uma aplicação *web* (escrita em *React.js*) numa móvel (escrita em *React Native*), e vice-versa.

3.3.1.1 Expo CLI

Framework que permite construir aplicações móveis em *React Native*. É o ideal para quem está a começar a conhecer esta tecnologia, visto que fornece várias ferramentas e serviços prontos a usar. Torna mais fácil a instalação de bibliotecas (o programador não tem que se preocupar com as diferentes instalações, para *iOS* e *Android*), bem como a construção final do ficheiro *APK*, para instalação em dispositivos *Android*. [45]

Com *Expo CLI*, é muito fácil preparar o ambiente de programação para a aplicação *React Native* a ser construída. Permite, também, testar a aplicação em qualquer tipo de sistema operativo (*iOS* e *Android*), apenas lendo o *QR code* que aparece no ecrã. No entanto, também apresenta limitações, como não ser possível adicionarem-se módulos nativos ou o facto de levar a aplicação a ocupar muito espaço, visto que esta é construída usando todas as bibliotecas integradas. [45]

3.3.1.2 React Native CLI

Funcionalidade integrada que permite ter controlo total sob a gestão do projeto, localmente. Apesar de poder ser mais complexo, quando comparado a *Expo CLI*, permite um desenvolvimento mais dedicado aos diferentes tipos de sistemas operativos (*iOS* e *Android*), visto que permite que sejam adicionados novos módulos em *Java* (linguagem usada para desenvolver aplicações *Android*) ou *Objective-C* (linguagem usada para desenvolver aplicações *iOS*). [45]

Uma possível desvantagem é o facto de ser preciso ter instalado *Android Studio* e *XCode*, para executar a aplicação e, se necessário, alterar certas configurações nela. No entanto, o *XCode* só se encontra disponível para *Mac*. Além disso, a preparação do projeto também pode levar algum tempo, sendo um processo que exige vários passos. [45]

3.3.2 FLUTTER

Framework de acesso aberto, desenvolvida pela *Google*, para construir aplicações multiplataforma e que compilam de forma nativa, a partir de um único código base [46].

Consiste em duas partes importantes [47]:

1. SDK (*Software Development Kit*): ferramentas que auxiliam a construção das aplicações, incluindo as necessárias para compilar o código escrito, para código nativo da máquina onde a aplicação vai correr. É de notar que a linguagem de programação usada em *Flutter* é *Dart*, uma linguagem orientada para *front-end*;
2. Biblioteca baseada em *widgets*: coleção de elementos reutilizáveis para a interface do utilizador (como botões e *inputs*), que podem ser personalizados.

3.3.3 ANDROID STUDIO

É o ambiente de desenvolvimento oficial para criação de aplicações para *Android*. É baseado em *Gradle*, suportando *C++* e *NDK*, e permite desenvolver aplicações para todos os dispositivos *Android*, apresentado emuladores ricos para testagem da aplicação [48].

No entanto, existem algumas desvantagens no desenvolvimento para *Android* [49]:

- Apesar da *Google* fazer atualizações constantes, devido aos ataques informáticos recorrentes a *Android*, a maioria dos utilizadores não tem por hábito atualizar o seu dispositivo regularmente. Devido a isto, muitas vezes, os desenvolvedores das aplicações têm que gerir questões relacionadas com segurança;
- Os fabricantes dos equipamentos criam os seus próprios *layouts*, pelo que uma aplicação ao ser executado em versões *Android* iguais, em dispositivos de modelos distintos, pode apresentar diferenças;
- São lançados muitos equipamentos, com propriedades diferentes (como tamanho de ecrã e *drivers* gráficos), minimizando a qualidade da aplicação, ao ser executado em equipamentos tão distintos;
- Apesar de ser relativamente fácil publicar-se uma aplicação na *Google Play Store*, a *Google* não garante direitos de autor nem verifica patentes – isto permite que sejam publicadas várias aplicações que têm as mesmas funcionalidades.

4 LEVANTAMENTO DE REQUISITOS

O presente capítulo é dedicado ao levantamento de requisitos, no contexto desta dissertação, analisando-se os funcionais e não funcionais. Será também realizada uma descrição dos perfis de utilização, associados ao sistema, e das respetivas personas, bem como a análise dos casos de uso do sistema.

4.1 CONTEXTO

Como tem vindo a ser exposto ao longo deste documento, o propósito desta dissertação passa por criar um sistema integrado de registo de propriedade, que reúna todas as funcionalidades e transações associadas a este registo predial, em São Tomé e Príncipe de modo a ser possível abandonar os livros em papel e passar a realizar todas as operações na forma digital.

Com este sistema, vai ser possível obter-se um *software* centralizado, que reúne todas as funcionalidades para o registo e outras operações relacionadas com prédios, terrenos, veículos, embarcações e aviões, bem como uma aplicação móvel que auxilia a pesquisa de informação sobre o registo automóvel.

Deve ainda ser mencionado que o projeto SIRP procura seguir, tal como previamente descrito, uma metodologia ágil, que envolve colaboração constante entre os membros do projeto (equipa de negócio e de desenvolvimento) e o trabalho por iterações (que podem durar semanas ou meses, mas tem-se preferência pelo menor intervalo de tempo possível). Esta metodologia opta por garantir que existe entrega frequente de valor, definindo o produto por etapas e assegurando assim que o mesmo vai de encontro ao que o cliente deseja (sendo a satisfação deste a prioridade principal) [50][51]. Assim, podem enumerar-se várias vantagens [50]:

- Faz com que equipa se dedique completamente a cada iteração;
- É muito eficiente, em termos de custo, para equipas com 25 desenvolvedores ou menos;
- Permite que os clientes vão moldando o projeto, à sua medida, conforme este se desenvolve;

- Garante facilidade em adaptar a solução às prioridades do cliente, havendo a garantia que os requisitos continuam a ser sempre relevantes;
- A confiança com o *stakeholder* é facilmente construída.

Desta forma, existem alguns requisitos associados que o sistema deve cumprir, por forma a funcionar como desejado, e alguns casos de uso que devem ser definidos.

4.2 REQUISITOS FUNCIONAIS

Através dos requisitos funcionais, é possível definirem-se as exigências do *stakeholder*, e que necessitam de estar presentes no sistema a ser desenvolvido, visto encontrarem-se assim acordadas com o mesmo. Vão, então, explicar aquilo que o sistema é capaz de fazer.

Os requisitos funcionais de um sistema devem [52]:

- Detalhar todas as operações presentes em cada ecrã;
- Tratar da lógica do tratamento de dados, a ser inserida no sistema;
- Descrever *outputs* do sistema;
- Detalhar informação acerca dos possíveis *workflows* do sistema;
- Definir, de forma clara, quem vai inserir, apagar e/ou modificar informação no sistema;
- Capturar como o sistema irá cumprir as necessidades acordadas.

Ora, os requisitos funcionais são, assim, aquilo que o desenvolvedor deve implementar por forma a permitir que o utilizador consiga realizar a ação que deseja [53], ao contrário dos requisitos não funcionais, que irão também ser analisados na secção 4.3, apresentada de seguida. Os requisitos funcionais necessários para uma correta construção do Sistema de Informação do Registo de Propriedade, de São Tomé e Príncipe, podem encontrar-se analisados ao detalhe na Tabela 2 e Tabela 3, mostradas abaixo.

Tabela 2 - Requisitos funcionais (aplicação web) e respetiva descrição

Requisito funcional	Descrição
Existência de perfis de utilização	<p>O sistema deve ter um conjunto de perfis de utilização de modo a condicionar as ações realizadas pelos seus utilizadores.</p> <p>Os perfis de utilização devem ter elevada granularidade e referir-se a conjuntos de ações específicas.</p> <p>Os utilizadores do sistema devem poder ter, cumulativamente, vários perfis de utilização de modo que possam realizar as ações adequadas ao seu perfil profissional na instituição.</p>
Registo de uma conta de utilizador	Para acesso ao sistema, cada utilizador deve ter a possibilidade de criar uma conta. Uma conta recém-criada não deve ter acesso a nenhuma funcionalidade relevante no sistema.
Início de sessão de trabalho	O sistema deve permitir que os utilizadores iniciem uma sessão de trabalho e que as funcionalidades disponibilizadas sejam compatíveis com o seu perfil de utilização.
Existência de um perfil de administrador	<p>O utilizador com o perfil de administrador deve permitir a criação de novos perfis de utilização e gerir os perfis de utilização dos demais utilizadores do sistema, adicionando ou retirando perfis de utilização. Deve ainda ser capaz de bloquear o acesso de utilizadores ao sistema.</p> <p>No entanto, o administrador apenas deve poder gerir utilizadores e não pode realizar/alterar registos prediais.</p>
Existência de um perfil de manutenção	O utilizador com o perfil de manutenção deve ser capaz de gerir uma máscara de catalogação, criando novas máscaras e/ou adicionando novos campos às máscaras existentes, indicando se são de preenchimento obrigatório ou opcional, alterando a sua ordem dentro da máscara.
Existência de um perfil de conservador	Um utilizador com este perfil deve poder bloquear registos onde tenha encontrado irregularidades.
Existência de um perfil de catalogação	O utilizador com perfil de catalogação deve poder preencher os <i>templates</i> com os registos já previamente existentes em formato de papel. Deve também ser capaz de averbar ou inscrever um registo.
Existência de um perfil de validação	Um utilizador com perfil de validação deve ser capaz de validar todos os registos que não tenham sido feitos pelo próprio.
Existência de um perfil de estatísticas	Um utilizador com perfil de estatísticas deve poder aceder a informações estatísticas (como, por exemplo, a quantidade de registos efetuados num mês, por utilizador).
Existência de um histórico de todos os registos	A base de dados do sistema deve armazenar todos os registos de propriedade, independentemente do seu tipo e do seu estado.

Armazenamento de pesquisas	Todas as pesquisas realizadas no sistema devem ser armazenadas numa tabela de histórico contendo informação sobre quem pesquisou e o que pesquisou.
Existência de um histórico de registos	Cada utilizador deve ser capaz de consultar o seu histórico de registos.
Troca de dados com outras entidades	O sistema deve permitir a troca de informação com outras entidades – autoridade tributária, habitação, DGRN, DGV, cadastro, etc.

Tabela 3 - Requisitos funcionais (aplicação móvel) e respetiva descrição

Requisito funcional	Descrição
Pesquisa por fotografia	A aplicação móvel deve permitir que o utilizador faça uma pesquisa por foto de matrícula.
Correção do texto retornado pelo reconhecimento de caracteres ótico	A aplicação móvel deve permitir que o utilizador possa editar o texto lido pelo sistema, que consta numa matrícula, caso este seja lido com erros.
Pesquisa por texto	A aplicação móvel deve permitir que o utilizador faça uma pesquisa por texto.
Leitura de QR Codes, presentes no Documento Único do Automóvel	A aplicação móvel deve permitir que o utilizador leia o QR Code presente no Documento Único do Automóvel, e que lhe seja mostrada a informação acerca desse mesmo veículo.
Assunção do modo escuro, quando ativo no dispositivo	A aplicação móvel deve ser capaz de assumir que o dispositivo em questão se encontra em modo escuro.
Aparecimento de uma mensagem de aviso, quando o dispositivo se encontra em modo escuro	A aplicação móvel deve mostrar uma mensagem de aviso ao utilizador, caso o dispositivo se encontre em modo escuro, perguntando-lhe se pretende continuar no modo escuro ou alterar para o modo claro. Esta mensagem só deve aparecer na primeira vez que o utilizador entra na aplicação com o dispositivo em modo escuro.
Alternância entre o modo claro e modo escuro	A aplicação móvel deve permitir que o utilizador alterne entre modo claro e modo escuro, sempre que assim o pretender.
Existência de um FAQ (<i>Frequently Asked Questions</i>)	A aplicação deve ter uma secção destinada a perguntas e respostas, onde o utilizador é capaz de perceber como a aplicação funciona.

4.3 REQUISITOS NÃO FUNCIONAIS

Ao contrário dos requisitos funcionais, os requisitos não funcionais definem os atributos de qualidade do sistema. Estes requisitos são usados para garantir robustez, ajudando a facilitar o *design* e o desenvolvimento do *software*, bem como verificar a *performance* do mesmo. Assim, os requisitos não funcionais são

definidos baseados em responsabilidade, usabilidade, segurança, portabilidade e outros critérios não funcionais que são críticos para o sucesso do sistema. [54]

Em suma, estes requisitos vão explicar como é que o sistema realmente funciona. Na Tabela 4, encontram-se reunidos os requisitos não funcionais necessários, no contexto desta dissertação, e que se aplicam tanto à aplicação *web* como móvel.

Tabela 4 - Requisitos não funcionais (aplicação *web* e móvel) e respetiva descrição

Requisito não funcional	Descrição
Mensagens de erro claras e específicas	As mensagens de erro apresentadas ao utilizador devem ser auto descritivas, na medida em que não devem deixar margem para diferentes interpretações acerca do erro que está a ser cometido.
Controlo de versões	Com um controlo de versões, tal como o próprio nome indica, é possível gerir diferentes versões de um projeto. Desta forma, os desenvolvedores envolvidos no projeto devem ser capazes de acompanhar um histórico da solução, podendo trabalhar paralelamente na mesma, apresentando as diferentes alterações que cada um realizou, e evitando assim duplicação ou sobreposição de código.
Resposta rápida a pedidos	Ao realizar um pedido ao sistema, o utilizador deve ter a resposta no mais curto espaço de tempo possível, garantindo-se assim um tempo de resposta mínimo.
Controlo de acessos indevidos	A segurança dos dados deve ser mantida, sendo que apenas utilizadores autenticados e autorizados nos perfis respetivos devem ser capazes de aceder à informação desejada.
Possibilidade de existência de utilizadores concorrentes	Os utilizadores devem poder aceder, de forma concorrente, à mesma informação, sem que isso traga nenhum tipo de problema ou interferência ao sistema.
Existência de responsividade	As aplicações <i>web</i> e móvel devem ser capazes de se adaptarem a qualquer tipo de ecrã, sem que isso interfira, de forma alguma, com a experiência do utilizador.
Possibilidade de usar a aplicação móvel em diferentes sistemas operativos	A aplicação móvel deve ser capaz de ser executada em sistemas operativos <i>Android</i> e <i>iOS</i> .

4.4 REQUISITOS RELATIVOS AO REGISTO AUTOMÓVEL

Ao contrário do que aconteceu no registo predial, onde se começaram por digitalizar os registos que já constavam em livros de formato físico, no registo automóvel o percurso seguido foi o do “*born digital*” – foi originado no formato

digital. Desta forma, ainda não existem quaisquer registos automóveis criados, para além daqueles que foram criados para propósito de teste.

Assim, foi também necessário um estudo sobre qual a informação necessária para criar um registo automóvel, bem como a averiguação de quais as entidades externas envolvidas no processo. Foram analisados vários documentos relativos ao registo automóvel, realizado em São Tomé e Príncipe e, a partir daqui, criado um esquema que esboça todo o processo.

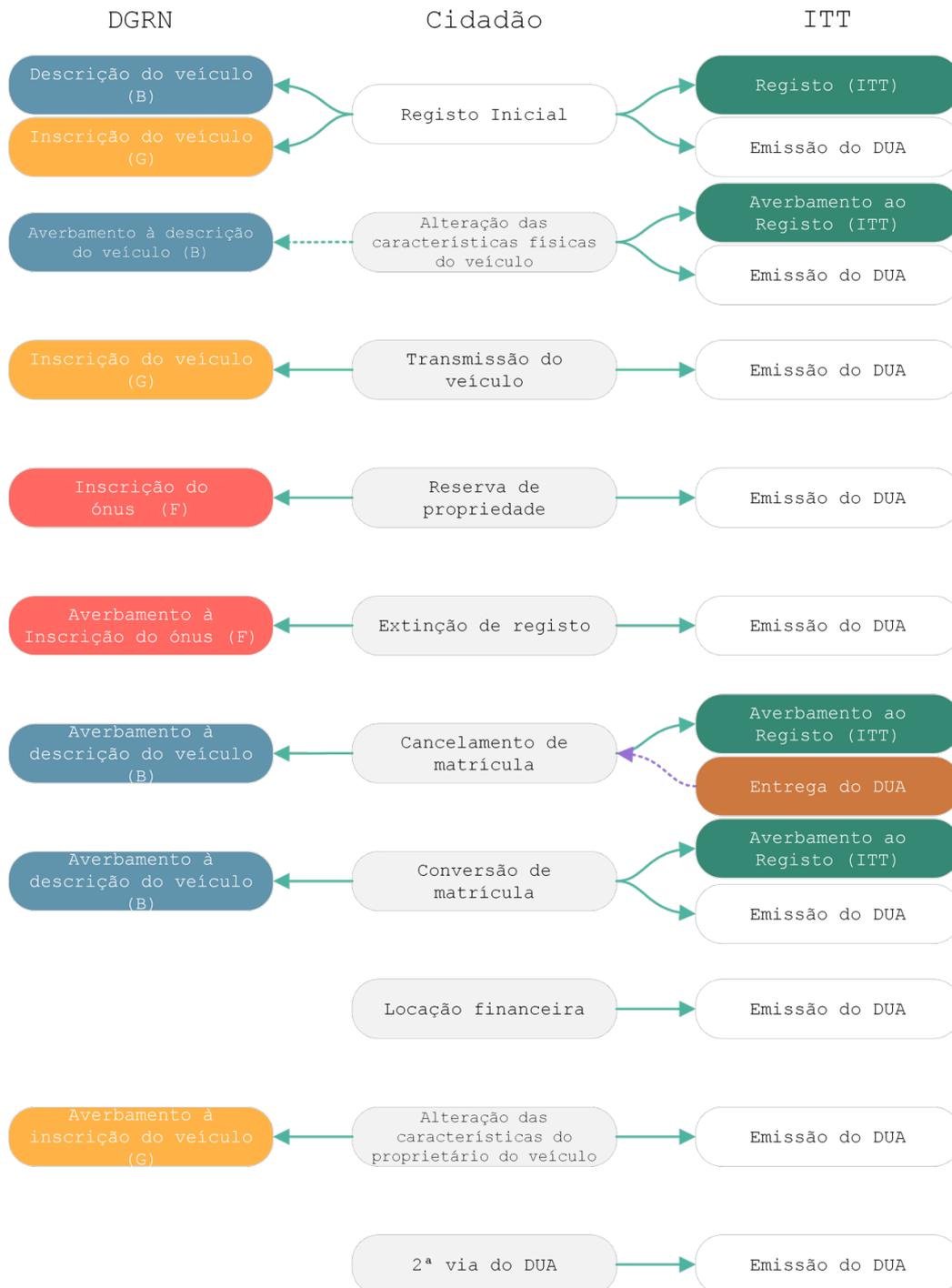


Figura 18 - Requisitos para a criação do registo automóvel

Foi assim possível averiguar-se que são três as entidades envolvidas no processo – DGRN, cidadão (que requer o registo) e ITT (Instituto dos Transportes Terrestres). Dependendo do tipo de ação que o cidadão deseja realizar, são desencadeados diferentes procedimentos pela DGRN e pelo ITT. Isto é necessário para a criação da interface, visto que terão que ser mostrados diferentes formulários ao utilizador, consoante o tipo de ação a realizar. É de notar que as letras que aparecem entre parêntesis na Figura 18 indicam qual o tipo de livro a que aquele registo pertence - B-Descrição, G-Inscrição da Propriedade, F-Inscrições Diversas, C-Inscrições Hipotecárias.

4.5 PERFIS DE UTILIZAÇÃO

Por forma a cumprirem-se os requisitos já apresentados, existem diferentes atores no sistema:

Tabela 5 - Perfis de utilização do sistema

Catálogo	<ul style="list-style-type: none"> • Cataloga as imagens e recolhe desta informação através das máscaras de catalogação.
Validação	<ul style="list-style-type: none"> • Valida registos catalogados no sistema. De modo a manter a transparência nos atos, a validação de um registo deve ser feita por um utilizador diferente do da catalogação.
Manutenção	<ul style="list-style-type: none"> • Cria/gere máscaras de catalogação – adiciona novos campos (de preenchimento obrigatório ou opcional) e altera a ordem dos mesmos; • Cria perfis de utilização e realiza a sua gestão; • Bloqueia e/ou dá acesso a utilizadores ao sistema; • Atribui/retira perfis de utilização dos utilizadores.
Conservador	<ul style="list-style-type: none"> • Bloqueia registos onde tenham sido encontradas irregularidades; • Desbloqueia registos, mediante inscrição de averbamento.
Administração	<ul style="list-style-type: none"> • Tem acesso a todos os históricos de registos e pesquisas realizados no sistema.
Pesquisa	<ul style="list-style-type: none"> • Permite que o utilizador possa efetuar pesquisas sobre registos que constam no sistema.
Estatísticas	<ul style="list-style-type: none"> • Consulta estatísticas globais relativas às ações realizadas no sistema.
Polícia	<ul style="list-style-type: none"> • Consulta informação relativa a matrículas de veículos, na aplicação móvel.

4.6 PERSONAS

A definição de *personas* é uma parte bastante importante do processo de levantamento de requisitos. Uma *persona*, sendo uma representação fictícia, define o tipo de pessoa que interage com o sistema – se se procura desenhar um *software* que cumpra as exigências do *stakeholder*, e que seja eficiente, deve procurar-se desenhá-lo para uma pessoa, ou conjunto de pessoas, em específico. Assim, conseguem identificar-se as vontades e necessidades do cliente, de forma bastante precisa, bem como entender as suas experiências, comportamentos e expectativas. [55][56]

Para o sistema em questão, e tendo por base os requisitos funcionais anteriormente mencionados, podem definir-se algumas *personas*. Estas encontram-se documentadas, com detalhe, no primeiro anexo deste documento.

4.7 USER STORIES

Um *user story* (US) passa por uma descrição escrita de uma funcionalidade do *software*, específica e orientada a um objetivo, normalmente constituída por uma única frase. Mostra a perspetiva do utilizador final, sendo que todos os *stakeholders* devem estar incluídos na sua elaboração. [57]

4.7.1 APLICAÇÃO WEB

Como mencionado acima, a solução final conta com cinco perfis de utilização, pelo que os *user stories* documentados nesta secção visam explicar quais as ações possíveis de serem realizadas por cada um deles.

Note-se que, relativamente ao registo automóvel, foram criados alguns diagramas de sequência, por forma a descrever todos os passos necessários para a realização de cada ação (e que seguem a lógica representada na Figura 18). Estes diagramas encontram-se presentes no Anexo 2 desta dissertação.

Administrador

- US 1** O administrador deve ser capaz de consultar um histórico relativo às ações de cada utilizador;
- US 2** O administrador deve ser capaz de consultar uma vista geral dos livros, tanto prediais como automóveis;
- US 3** O administrador deve ser capaz de consultar as catalogações num dado intervalo de tempo, numa vista geral para todos os catalogadores ou especificando um;

- US 4 O administrador deve ser capaz de consultar um histórico de imagens;
- US 5 O administrador deve ser capaz de consultar um histórico de registos;
- US 6 O administrador deve ser capaz de consultar estatísticas;
- US 7 O administrador deve ser capaz de consultar um histórico de pesquisas realizadas na plataforma, por utilizador e tipo de dispositivo (computador/dispositivo móvel).

Mantenedor

- US 1 O mantenedor deve ser capaz de gerir os utilizadores da plataforma, bem como os dados de um utilizador - palavras-passe, *e-mail*, *username* e nome completo;
- US 2 O mantenedor deve ser capaz de conceder ou remover o acesso à plataforma, a cada utilizador;
- US 3 O mantenedor deve ser capaz de adicionar e/ou remover os papéis (perfis de utilização) associados a cada utilizador;
- US 4 O mantenedor deve ser capaz de consultar quais os papéis (perfis de utilização) associados a cada utilizador;
- US 5 O mantenedor deve ser capaz de consultar uma lista relativa aos papéis (tipos de perfis) existentes no sistema;
- US 6 O mantenedor deve ser capaz de adicionar novos papéis (tipos de perfis) ao sistema;
- US 7 O mantenedor deve ser capaz de consultar quais os utilizadores associados a cada tipo de papel (tipo de perfil);
- US 8 O mantenedor deve ser capaz de editar o nome de cada tipo de papel (tipo de perfil);
- US 9 O mantenedor deve ser capaz de consultar uma lista relativa aos modelos de dados;
- US 10 O mantenedor deve ser capaz de editar um modelo de dados;
- US 11 O mantenedor deve ser capaz de visualizar os detalhes de um modelo de dados;
- US 12 O mantenedor deve ser capaz de eliminar um modelo de dados, caso ainda não tenham sido realizados registos usando mesmo;
- US 13 O mantenedor deve ser capaz de testar cada modelo de dados, por forma a garantir o correto funcionamento do mesmo;

- US 14 O mantenedor deve ser capaz de consultar uma lista relativa ao tipo de modelos de dados existentes;
- US 15 O mantenedor deve ser capaz de criar um novo tipo de modelo de dados;
- US 16 O mantenedor deve ser capaz de editar um tipo de modelo de dados existente;
- US 17 O mantenedor deve ser capaz de apagar um tipo de modelo de dados existente;
- US 18 O mantenedor deve ser capaz de consultar uma lista relativa aos campos existentes;
- US 19 O mantenedor deve ser capaz de criar um novo campo;
- US 20 O mantenedor deve ser capaz de editar um campo existente;
- US 21 O mantenedor deve ser capaz de eliminar um campo existente;
- US 22 O mantenedor deve ser capaz de consultar uma lista relativa aos estados civis existentes na plataforma;
- US 23 O mantenedor deve ser capaz de criar um novo estado civil;
- US 24 O mantenedor deve ser capaz de editar um estado civil existente;
- US 25 O mantenedor deve ser capaz de eliminar um estado civil existente.

Catalogador

- US 1 O catalogador deve ser capaz de consultar a lista de imagens dos livros, podendo nela realizar uma pesquisa filtrada;
- US 2 O catalogador deve ser capaz de consultar informação acerca dos registos realizados por si próprio, podendo aí realizar uma pesquisa filtrada;
- US 3 O catalogador deve ser capaz de consultar estatísticas sobre os registos realizados por si próprio.

Conservador e validador

- US 1 O conservador/validador deve ser capaz de realizar as mesmas ações que um catalogador;
- US 2 O conservador/validador deve ser capaz de bloquear e desbloquear registos;
- US 3 O conservador/validador deve ser capaz de editar e validar registos.

- US 4** O conservador/validador deve ser capaz de criar um livro único, relativo ao registo predial.

Perfil de estatísticas

- US 1** O perfil de estatísticas deve ser capaz de consultar estatísticas relativas ao histórico de ações efetuadas por cada utilizador.

Gerais a todos os utilizadores

Caso de uso gerais, que se aplicam a todos os tipos de utilizadores mencionados acima.

- US 1** Enquanto utilizador, deve ser possível o registo na plataforma;
- US 2** Enquanto utilizador, deve ser possível o *login* na solução.

4.7.2 APLICAÇÃO MÓVEL

Na aplicação móvel, apenas existe um tipo de utilizador. Assim:

- US 1** Enquanto utilizador, deve ser possível a pesquisa por matrícula – por texto, carregando uma imagem da galeria ou digitalizando uma matrícula através da câmara do dispositivo;
- US 2** Enquanto utilizador, deve ser possível realizar a leitura do *QRCode* presente no documento único do veículo;
- US 3** Enquanto utilizador, deve ser possível saber detalhes sobre o veículo associado a uma matrícula, caso ela exista na base de dados;
- US 4** Enquanto utilizador, deve ser possível alternar entre modo escuro e modo claro, na aplicação.

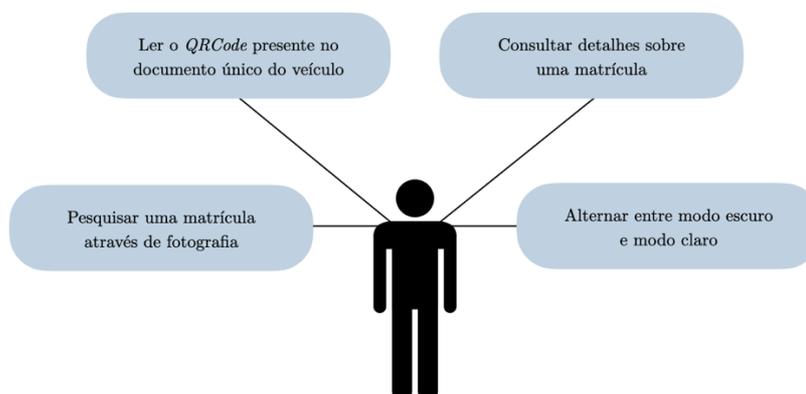


Figura 19 - Diagrama de casos de uso (aplicação móvel)

5 IMPLEMENTAÇÃO DA SOLUÇÃO

No capítulo “Implementação da Solução”, serão apresentados diversos aspetos que devem ser tidos em conta na conceção da solução, tendo sempre em conta os requisitos já definidos no Capítulo 4 deste documento.

5.1 ESCOLHA DAS TECNOLOGIAS

Como mencionado anteriormente neste documento, esta dissertação surgiu como resultado de uma proposta de bolsa de investigação. Assim, aquando da realização desta dissertação, já existiam componentes e funcionalidades previamente implementadas, no âmbito deste projeto. Desta forma, algumas tecnologias já haviam sido escolhidas:

- *Framework ASP.NET*, da *Microsoft*, para desenvolvimento de *backoffice*, *frontoffice* e *API*;
- Base de dados *SQL Server*, da *Microsoft*, para suportar a estrutura de dados.

Já a aplicação móvel, elaborada após o começo desta dissertação, foi desenvolvida em *React Native CLI*. Esta escolha foi feita tendo em conta que esta dissertação seria desenvolvida com o auxílio de um computador *Mac*, permitindo assim a existência das condições necessárias para programar, também, uma aplicação em *iOS*. Desta forma, será possível ter-se uma aplicação 100% funcional, tanto para *iOS* como para *Android*, visto existirem condições para a testar em ambos os sistemas operativos.

5.2 ARQUITETURA DO SISTEMA

De acordo com o descrito, surge a seguinte arquitetura aplicacional:

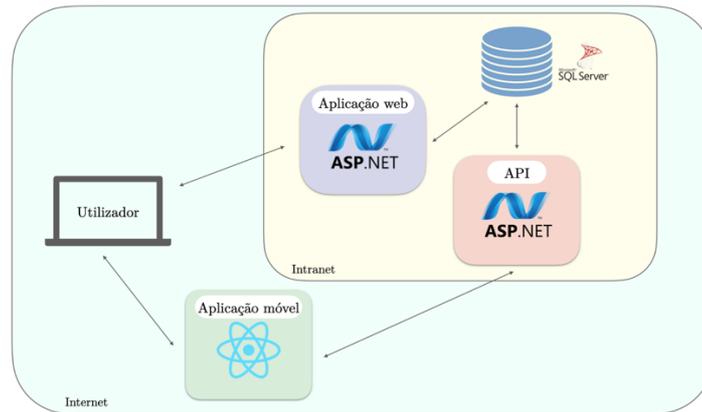


Figura 20 - Arquitetura do sistema

Como se pode observar na Figura 20, a arquitetura é composta por:

- Base de dados, desenvolvida em *SQL Server*, onde consta toda a informação usada na solução;
- Aplicação *web*, desenvolvida com recurso a *ASP.NET*, na linguagem de programação *C#*, que reúne as funcionalidades necessárias ao registo de propriedade e transações associadas ao ciclo de vida da mesma;
- *API*, desenvolvida com recurso a *ASP.NET*, na linguagem de programação *C#*, usada para responder aos pedidos das aplicações remotas;
- Aplicação móvel, desenvolvida com recurso a *React Native CLI*, que permite que os utilizadores da mesma façam pesquisas por matrículas automóveis e recebam, como resposta, informação relativa a esse veículo.

É de notar que as interações entre as aplicações e serviços estão representadas pelas setas da Figura 20.

5.3 ESTRUTURA DA BASE DE DADOS

Como mencionado, quando do começo desta dissertação, alguns componentes já haviam sido implementados - tal aconteceu com a base de dados. No entanto, visto estar sempre suscetível a mudanças, já sofreu alterações desde então (nomeadamente, no formato das imagens).

Existem algumas tabelas que são criadas automaticamente, pela *framework*:

- *AspNetUsers*, onde constam os utilizadores registados na plataforma e que têm acesso à mesma;
- *AspNetRoles*, onde constam os perfis de utilização do sistema;

- *AspNetUserRoles*, onde estão associados utilizadores a diferentes perfis de utilização;
- *AspNetUserLogins*, onde constam *logins* efetuados através de serviços externos, como por exemplo através do Facebook. Esta funcionalidade ainda não foi implementada, no SIRP;
- *AspNetUserClaims*, onde constam os utilizadores que realizaram login através de serviços externos.

A análise das tabelas presentes na base de dados, bem como o esquema da mesma, encontram-se descritos ao detalhe no Anexo 3 deste documento.

5.4 APLICAÇÃO WEB

Tal como anteriormente mencionado, a aplicação *web* foi desenvolvida com recurso a *ASP.NET*, na linguagem de programação *C#*. Além disso, foi usada a biblioteca *Bootstrap*¹⁵, que permite a reutilização de componentes e classes *css*, totalmente responsivas, adaptando-se às diferentes resoluções de ecrã.

Como referido neste documento, aquando do começo desta dissertação, já existiam algumas partes do sistema implementadas, dado o âmbito em que a mesma está inserida - tal aconteceu com a aplicação *web*.

5.4.1 REGISTO PREDIAL

A parte relativa ao registo predial era, então, aquela que já se encontrava em desenvolvimento aquando do começo desta dissertação. No Anexo 4 deste documento, encontram-se algumas imagens referentes ao registo predial. Note-se que, aqui, todos os livros que constavam em formato físico foram digitalizados, encontrando-se agora a sofrer a respetiva catalogação (também no formato digital).

A única mudança realizada na aplicação *web*, no âmbito desta dissertação, foi o formato das imagens pertencentes aos livros físicos, relativos ao registo predial (alteração feita na base de dados, mas que se reflete na *interface*). Inicialmente, era seguido um formato do género “PG-2017-INP-027-001”, onde:

- “PG” indicava o livro fonte, onde a imagem pertence;
- “2017” indicava o ano do livro – ou seja, o ano em que foram realizados os registos pertencentes àquele livro;

¹⁵ <https://getbootstrap.com> (Acedido em 22 de julho de 2022)

- “INP” era usado como acrónimo – neste caso, INP significava “inscrição predial”;
- “027” indicava o número do livro;
- “001” indicava o número da página.

Visto que alguns campos acabavam por nunca ser usados, o formato das imagens tornou-se algo mais autoexplicativo, seguindo-se um formato do género “PG-027-001-F”, onde:

- “PG” indica o livro fonte ao qual a imagem pertence;
- “027” indica o número desse mesmo livro;
- “001” indica o número da folha;
- “F” indica que é a “frente” da folha (este campo pode também tomar o valor “V”, caso se trate do verso”).

5.4.2 REGISTO AUTOMÓVEL

Como já mencionado, ao contrário do registo predial, que já havia sido implementado, o registo automóvel foi implementado durante o desenvolvimento desta dissertação. Assim, a sua implementação será explorada mais ao pormenor, nesta secção. Note-se que a interface criada para o registo automóvel se encontra, maioritariamente, no Anexo 5 desta dissertação.

Máscaras de catalogação

As máscaras de catalogação (*templates*) permitem ter diferentes formulários para diferentes tipos de registo, sendo que os campos diferem de máscara para máscara. Estas já existiam, antes da implementação do registo automóvel.

Tipo de prop.	Nome do modelo	Tipo de modelo	Campos	Registos
Automóvel	Livro Único Averbamento à Descrição	Averbamento	29	0
	Livro Único Averbamento à Inscrição	Averbamento	6	1
	Livro Único Averbamento à Inscrição do Ónus	Averbamento	7	0
	Livro Único Averbamento ao Registo	Transportes	29	1
	Livro Único Descrição	Descrição	8	1
	Livro Único Inscrição	Inscrição	25	2
	Livro Único Inscrição do Ónus	Inscrição	11	0
	Livro Único Registo	Transportes	9	1
Predial	Livro em papel Averbamento à Descrição	Averbamento	22	214
	Livro em papel Averbamento à Inscrição	Averbamento	10	4
	Livro em papel Descrição	Descrição	23	1065
	Livro em papel Inscrição	Inscrição	13	31
	Livro Único Descrição	Descrição	21	0
	Livro Único Inscrição	Inscrição	2	0

Figura 21 - Página para consulta de máscaras de catalogação

Como se pode observar pela Figura 61 do Anexo 5, é possível adicionar novos campos, a qualquer momento, a uma máscara de catalogação, bem como alterar a ordem dos mesmos (através da *interface*). Isto permite que os formulários sejam dinâmicos e tenham sempre os campos necessários (campos estes que também podem ser criados diretamente na interface).

Criação de um registo

Como mencionado no Capítulo 4 deste documento, a criação de um registo automóvel implica a intervenção de duas entidades externas – a DGRN e o ITT.

Ora, para a criação de um registo automóvel inicial, na plataforma, são então precisos três formulários – descrição e inscrição do veículo, com a DGRN, e registo do veículo, com o ITT. Estes três registos são guardados simultaneamente na base de dados, e mostrados ao utilizador através de secções (Figura 63 do Anexo 5). O número de cada registo é automaticamente gerado – de momento, é atribuído um número aleatório a cada um.

Se, porventura, o utilizador tentar criar um registo com campos de preenchimento obrigatório vazios, recebe um aviso e só consegue continuar a sua ação depois de preencher todos os campos em falta.

Livro de Registos

Nesta página é possível realizar consultas aos registos existentes. Nela, também se podem encontrar várias outras funcionalidades, relacionadas com o registo automóvel – criação, edição, visualização, eliminação, validação, criação de averbamentos e inscrições, criação do ficheiro relativo à nota de

lançamento de cada registo e emissão do Documento Único do Automóvel (esta última funcionalidade ainda não está implementada).

A pesquisa pode ser realizada por tipo de livro, estado de registo ou número de registo.

Edição de registos

Para se efetuar uma edição de um registo, é necessário o cuidado de se verificar se esse registo é um registo inicial (ou seja, tem mais dois registos associados), ou se é um registo singular (averbamentos, inscrições ou descrições), criado sobre o registo inicial de um veículo.

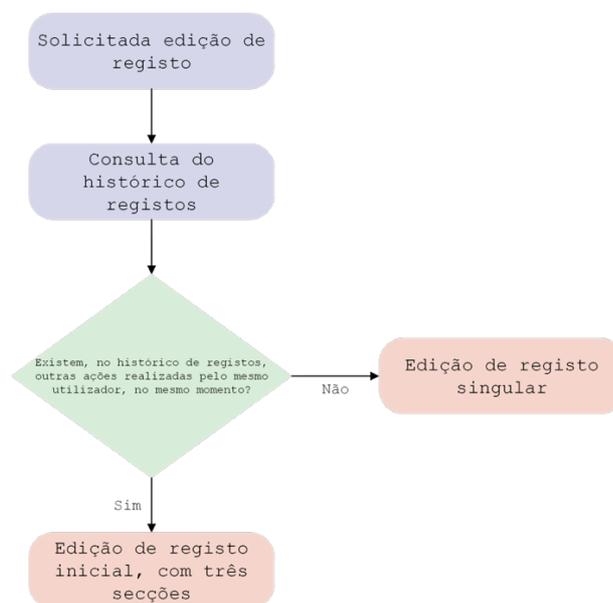


Figura 22 - Fluxograma do processo de edição de um registo

Como se pode observar na Figura 22, existem dois caminhos possíveis, quando se pretende editar um registo:

1. Caso se encontrem ações, no histórico de registos, realizadas noutros registos, pelo mesmo utilizador, no mesmo momento, significa que foram criados mais do que um registo, simultaneamente. Isso é, desta forma, indicativo de que se trata de um registo automóvel inicial, pelo que o utilizador deve conseguir editar os três registos associados. São-lhe assim mostradas, na *interface*, três secções distintas, para poder editar os três registos, como se pode observar na Figura 65 do Anexo 5;
2. Caso não exista mais nenhuma ação realizada naquele momento, pelo mesmo utilizador, significa que o utilizador selecionou um registo singular, pelo que lhe é mostrado, na *interface*, uma página única de edição, como se pode observar na Figura 66 do Anexo 5.

Caso o utilizador tente validar/invalidar um registo com campos de preenchimento obrigatório vazios, recebe um aviso e só consegue continuar a sua ação depois de preencher todos os campos em falta.

Consulta de detalhes de registos

Na página de visualização de detalhes de um registo, é possível verificarem-se quais os campos preenchidos (e com que informação), bem como se existem registos dependentes do registo selecionado, o estado do registo e o histórico de ações realizadas sobre o mesmo (Figura 68 do Anexo 5).

Averbar/inscrever um registo

Um registo pode ser averbado caso seja do tipo “Inscrição”. Assim, na página *web* do livro de registos, todas as inscrições contêm um botão para realizar o averbamento (Figura 64 do Anexo 5). Após o utilizador clicar nesse botão, pode escolher o tipo de averbamento que deseja fazer (Figura 69 do Anexo 5).

Um averbamento tem um número de registo específico – pode ser do formato “número de registo original/1”, caso seja o primeiro averbamento daquele registo, ou “número de registo original/(contagem do número de averbamentos+1)”. Este processo é apresentado através do fluxograma da Figura 23. Suponha-se que o número de registo a averbar é o 144 – se este ainda não contiver registos, o averbamento em causa terá, como número de registo, “144/1”, visto ser o primeiro averbamento. Caso o registo 144 já contenha, por exemplo, cinco averbamentos, o averbamento em causa terá como número de registo “144/6”, visto ser o sexto averbamento existente, relativo ao registo que se está a averbar.

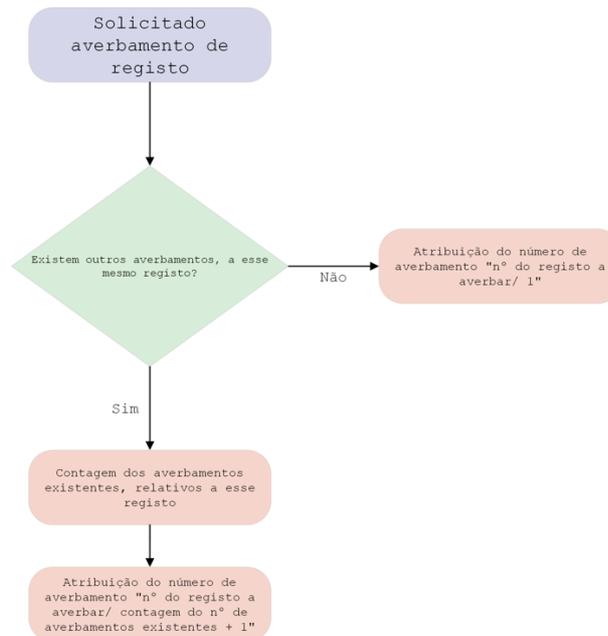


Figura 23 - Fluxograma relativo à criação de um averbamento

Relativamente às inscrições, um registo pode ser inscrito caso seja do tipo “Descrição”. Desta forma, na página *web* do livro de registos, todas as descrições contêm um botão para realizar a respetiva inscrição (Figura 64 do Anexo 5). Após o utilizador clicar nesse botão, pode escolher o tipo de inscrição que deseja fazer (Figura 67 do Anexo 5).

Validar/invalidar um registo

Caso um registo esteja no estado “catalogado”, é mostrado ao utilizador um botão para validar o registo. Após clicar nele, pode validar ou invalidar o registo (Figura 70 do Anexo 5). O estado de um registo é sempre apresentado na página relativa ao livro de registos. Se um registo, previamente validado (Figura 71 do Anexo 5), for editado, ele passa para o estado “editado” (Figura 72), e requer uma nova validação. Novamente se, porventura, o utilizador tentar validar/invalidar um registo com campos de preenchimento obrigatório vazios, recebe um aviso e só consegue continuar a sua ação depois de preencher todos os campos em falta.

Eliminar um registo

De momento, apenas é possível eliminar registos singulares. Ainda não é possível eliminar um registo inicial (que contém três registos) pois ainda não foi acordado, no âmbito do projeto, se devem ser eliminados em simultâneo ou separadamente. A página para eliminar um registo singular é mostrada na Figura 73 do Anexo 5. Além disso, caso um registo contenha averbamentos, também não poderá ser apagado.

5.5 APLICAÇÃO MÓVEL – SIRP AUTO

Como já referido, a aplicação móvel, de nome SIRP Auto, foi desenvolvida com auxílio a *React Native CLI*. Desta forma, durante o desenvolvimento da mesma, foram usados diferentes dispositivos para *debug*:

- *Tablet Samsung Galaxy Tab A8 SM-X200* (com sistema operativo *Android*);
- *Smartphone Xiaomi Redmi Note 8T* (com sistema operativo *Android*);
- Emulador de *iPhone 13* (com sistema operativo *iOS*).

A existência de um leque de dispositivos variados, que diferem tanto em sistema operativo como em dimensões, permitiu que, durante o desenvolvimento da aplicação, se garantisse que a mesma seria capaz de se adaptar a qualquer equipamento *iOS* ou *Android*. É de notar que não foi possível testar a aplicação num dispositivo físico *iOS*, visto que para isso é necessária a licença de desenvolvedor da *Apple*. No entanto, dado que foi usado o emulador e que existia acesso ao ambiente de desenvolvimento *XCode*, através do *Mac*, é na mesma garantida a qualidade da aplicação nesse sistema operativo.

O maior desafio aqui encontrado foi, sem dúvida, o facto de se estar a desenvolver, simultaneamente, uma aplicação apta a ser executada em dois sistemas operativos distintos. Além disso, o *Mac* usado tem o processador *M1* da *Apple*, sendo que algumas especificações do *React Native CLI* não eram compatíveis (no momento) com esta arquitetura, o que levou a que o processo de preparação do ambiente de programação fosse mais demorado.

5.5.1 PROPÓSITO DA APLICAÇÃO

Com o desenvolvimento da aplicação SIRP Auto pretende-se, então, que seja possível pesquisar-se por uma matrícula e obter informação oficial acerca daquele veículo.

Dado que a SIRP Auto vai ser usada por entidades policiais, para consultarem informação sempre que necessário, é preciso que a pesquisa seja fácil de realizar, e que não demore muito tempo. Assim, além da clássica pesquisa escrita, por matrícula, pretende-se que também seja possível usar a câmara do dispositivo, fazendo-se assim uma leitura rápida da matrícula, ou usando-se uma imagem da galeria. Estas duas últimas opções podem parecer redundantes, mas o uso de uma foto da galeria, tirada anteriormente, pode ser útil. É de notar que em São Tomé existem vários locais sem *internet*. Ora, desta forma, o polícia pode tirar uma foto à matrícula que deseja pesquisar

(sem necessitar de estar a perder tempo a escrever) e, quando tiver uma ligação à *internet* disponível, basta carregar a foto na aplicação.

Além desta abordagem, vai também ser possível ler-se os *QRCode* presentes no Documento Único do Automóvel, de São Tomé, dos diversos veículos. Mais uma vez, esta técnica irá poupar tempo aos polícias, visto que basta apontarem a câmara para o *QRCode* presente no documento e receberão, em retorno, toda a informação acerca do veículo. Será também útil aquando da realização de operações de fiscalização, no país.

Uma vez que a API ainda não foi construída no âmbito do projeto, foi usado um ficheiro *JSON* (Figura 24) para simular os dados, na aplicação móvel. Assim, é possível ter-se uma noção mais completa do resultado final da aplicação, quando esta começar a ser utilizada em contexto real.

```
{
  "info": [
    {
      "id": 1235443233,
      "sujeitoAtivo": {
        "nomeCompleto": "João Costa",
        "bilheteIdentidade": null,
        "numeroPassaporte": 1234567,
        "maior": true,
        "emancipado": null,
        "estadoCivil": "Solteiro",
        "residencia": "São Tomé",
        "codigoPostal": "0000-000",
        "conjugue": null,
        "nomeConjugue": null,
        "regimeBens": null
      },
      "sujeitoPassivo": null,
      "caracteristicasFisicas": {
        "matricula": "STP-03-78U",
        "dataMatricula": "05-12-2013",
        "marca": "Toyota",
        "modelo": "RAV4",
        "categoria": "D",
        "tipo": "Automóvel",
        "caixa": {
          "dimensoes": "100x100",
          "tipo": "Manual"
        }
      },
      "caracteristicasTecnicas": {
        "anoFabrico": "2010",
        "VIN": "WP0ZZZ99ZTS392124",
        "motor": {
          "numeroMotor": "3764857839AKF334",
          "numeroCilindros": 3,
          "cilindrada": "1000 cm3",
          "combustivel": "Gasolina 98"
        },
        "medidasPneumaticos": {
          "frente": "195 mm",
          "retaguarda": "195 mm"
        },
        "lotacao": 5,
        "pesos": {
          "bruto": "3500 kg",
          "maximos": {
            "frente": "300 kg",
            "retaguarda": "400 kg"
          }
        },
        "DEE": "12345AJ",
        "tara": "3000 kg"
      }
    }
  ]
}
```

Figura 24 – JSON com informação emulada, relativa a uma matrícula

5.5.2 ESPECIFICAÇÕES INICIAIS

Como já mencionado, a preparação do ambiente de programação com *React Native CLI* é mais penoso, tanto a nível de tempo como a nível de memória do computador. Para desenvolvimento através de um *Mac*, é necessário:

- Instalação do *Node* e *Watchman*;
- Instalação da distribuição do *OpenJDK*, *Azul Zulu*;
- Configuração das variáveis de ambiente de *Android*;
- Instalação do *Android Studio* e do *XCode*;
- Transferência de um emulador de *iOS*, através de *XCode*.

Após reunidas as condições, é possível criar-se um projeto *React Native CLI*. Nele, vêm já criadas os diretórios “*ios*” e “*android*”, com código *Java* e *Objective-C*, respetivamente.

5.5.3 RESPONSABILIDADE DA APLICAÇÃO

A responsividade da aplicação é um aspeto bastante importante a ter em conta, ao longo do desenvolvimento da mesma, visto que a SIRP Auto deve estar preparada para ser usada, de forma eficiente e correta, em diferentes dispositivos de diferentes dimensões e com sistemas operativos distintos.

Bibliotecas utilizadas

Duas bibliotecas ganham destaque, neste contexto:

- ***React Native Safe Area Context*¹⁶**: usada com o propósito de assegurar que o *layout* da aplicação fica dentro das margens, em *iOS*. Permite aplicar *padding* automático, assegurando que as margens arredondadas dos dispositivos *iOS* ou a área de sensores e câmara não afetem o *layout*;
- ***React Native Orientation Locker*¹⁷**: permite saber qual a orientação do ecrã (horizontal ou vertical), bem como bloquear a aplicação numa certa orientação. Desta forma, é possível adaptar certos aspetos do *layout*, como o tamanho de botões ou do leitor de *QRCode*, que sem este cuidado ficariam desproporcionais ao mudar de orientação.

5.5.4 ACESSO À APLICAÇÃO – TOKEN DE AUTENTICAÇÃO

Apesar de a API ainda não ter sido desenvolvida, a aplicação móvel foi desenvolvida tendo em conta que, para que seja possível efetuarem-se pedidos à mesma, será necessária a aquisição de um *token* de autenticação. De momento, esse *token* está definido diretamente no código, como uma constante.

Visto que a aplicação não terá *login*, foi criado um ecrã inicial (Figura 25) que permite simular essa ação. Nele, consta um botão “Entrar” – quando o utilizador carrega neste, serão enviados à API (quando a mesma existir) um *e-mail* e *password* padrão (definidos aquando da criação da API), que

¹⁶ <https://github.com/th3rdwave/react-native-safe-area-context> (Acedido em 22 de julho de 2022)

¹⁷ <https://github.com/wonday/react-native-orientation-locker> (Acedido em 22 de julho de 2022)

permitem simular o início de sessão. Desta forma, consegue obter-se o *token* necessário para realizar pedidos à API, durante a utilização da aplicação.



Figura 25 - Página inicial da aplicação (Android e iOS, respetivamente)

5.5.5 RECONHECIMENTO ÓTICO DE CARACTERES

Reconhecimento Ótico de Caracteres – em inglês, *Optical Character Recognition* (OCR) - é uma técnica que permite extrair informação de documentos e/ou imagens, convertendo o texto para uma forma em que possa ser processado por uma máquina, como por exemplo em pesquisas [58].

Por forma a ser possível lerem-se as matrículas dos veículos, a partir da câmara do dispositivo ou de uma fotografia previamente tirada, é então necessário ter-se um sistema de reconhecimento ótico de caracteres.

Acesso à câmara do dispositivo

Para que seja permitido usar-se a câmara do dispositivo, é necessário pedir-se autorização ao utilizador, para que esse acesso seja concebido. Assim, deve ter-se em atenção este pormenor, tanto para *Android* como para *iOS*, e adicionar as configurações necessárias.

Em *Android*, deve incluir-se a permissão no ficheiro *AndroidManifest.xml*, que contém toda a informação sobre o projeto *Android*, incluindo componentes usados.

```
AndroidManifest.xml X
android > app > src > main > AndroidManifest.xml
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2   package="com.appautomovel">
3
4   <uses-permission android:name="android.permission.INTERNET" />
5   <uses-permission android:name="android.permission.CAMERA" />
6   <uses-feature android:name="android.hardware.camera" android:required="false" />
7   <uses-feature android:name="android.hardware.camera.front" android:required="false" />
8   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
9
10  <application
11    android:name=".MainApplication"
12    android:label="@string/app_name"
13    android:icon="@mipmap/ic_launcher"
14    android:roundIcon="@mipmap/ic_launcher_round"
15    android:allowBackup="false"
16    android:theme="@style/AppTheme">
17    <activity
18      android:name=".MainActivity"
19      android:label="@string/app_name"
20      android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|screenSize|smallestScreenSize|uiMode"
21      android:launchMode="singleTask"
22      android:windowSoftInputMode="adjustResize"
23      android:exported="true">
24      <intent-filter>
25        <action android:name="android.intent.action.MAIN" />
26        <category android:name="android.intent.category.LAUNCHER" />
27      </intent-filter>

```

Figura 26 - Permissões necessárias para ser possível aceder à câmara, em Android

Já em *iOS*, o ficheiro a ser alterado é o *Info.plist*, visto que é ele o responsável por configurar a aplicação.

```
Info.plist X
ios > AppAutomovel > Info.plist
29 <key>NSExceptionDomains</key>
30 <dict>
31   <key>localhost</key>
32   <dict>
33     <key>NSExceptionAllowsInsecureHTTPLoads</key>
34     <true/>
35   </dict>
36 </dict>
37 </dict>
38 <key>NSLocationWhenInUseUsageDescription</key>
39 <string>Location access is required to run the app.</string>
40 <key>NSPhotoLibraryUsageDescription</key>
41 <string>Photo Library access is required to run the app.</string>
42 <key>NSCameraUsageDescription</key>
43 <string>Camera access is required to run the app.</string>
44 <key>UILaunchStoryboardName</key>
45 <string>LaunchScreen</string>
46 <key>UIRequiredDeviceCapabilities</key>
47 <array>
48   <string>armv7</string>
49 </array>
50 <key>UISupportedInterfaceOrientations</key>
51 <array>
52   <string>UIInterfaceOrientationPortrait</string>
53   <string>UIInterfaceOrientationLandscapeLeft</string>
54   <string>UIInterfaceOrientationLandscapeRight</string>
55 </array>

```

Figura 27 - Permissões necessárias para ser possível aceder à câmara, em iOS

Bibliotecas utilizadas

Após garantido o acesso à câmara, escolheram-se as bibliotecas que permitem o reconhecimento ótico de caracteres. Foram necessárias duas:

- ***React Native Image Crop Picker***¹⁸: compatível com *iOS* e *Android*, permite seleccionar imagens da galeria, usar a câmara para tirar fotografias e recortar as mesmas;
- ***React Native MLKit OCR***¹⁹: *ML Kit* da *Google*, otimizado para dispositivos móveis, que permite aplicar *machine learning* em aplicações *iOS* e *Android*. Neste caso, a biblioteca é orientada para o reconhecimento e leitura de texto presente em imagens.

Desta forma, com a biblioteca *React Native Image Crop Picker* é possível seleccionar-se uma foto que, seguidamente, é lida pela biblioteca de *machine learning* e o reconhecimento ótico de caracteres é conseguido.

```
const takePhotoFromCamera = () => {
  setShowOptions(false);
  setShowFlatList(true);
  ImagePicker.openCamera({
    compressImageMaxWidth: 300,
    compressImageMaxHeight: 300,
    cropping: true,
    compressImageQuality: 0.7
  }).then(image => {
    console.log(image);
    setImage(image.path);
    //this.bs.current.snapTo(1);
    getText(image.path);
  })
  .catch(error => {
    if (error.code === 'E_PICKER_CANCELLED') { // handle error if user cancels
      return false;
    }
  });
};

const choosePhotoFromLibrary = () => {
  setShowOptions(false);
  setShowFlatList(true);
  ImagePicker.openPicker({
    width: 300,
    height: 300,
    cropping: true,
    compressImageQuality: 0.7
  }).then(image => {
    console.log(image);
    setImage(image.path);
    //this.bs.current.snapTo(1);
    getText(image.path);
  })
  .catch(error => {
    if (error.code === 'E_PICKER_CANCELLED') { // handle error if user cancels
      return false;
    }
  });
};
```

Figura 28 - Funções para leitura de imagem, através do uso da biblioteca "React Native Image Crop Picker"

¹⁸ <https://github.com/ivpusic/react-native-image-crop-picker> (Acedido em 22 de julho de 2022)

¹⁹ <https://github.com/agoldis/react-native-mlkit-ocr> (Acedido em 22 de julho de 2022)

```
const getText = async (path) => {  
  const result = await MLkitOcr.detectFromFile(path);  
  let newTmp = "";  
  if(result.length > 0){  
  
    let tmp = result[0].text.toUpperCase().replace(/^[^a-zA-Z0-9]+/g, "");  
    const tmpLen = tmp.length  
  
    /* String manipulation */  
  
    if(tmp.substring(0,4).includes("STP") && tmp.charAt(0) != "S"){ /* In case some character, in the first position, was  
      let index = tmp.indexOf("STP")  
      newTmp = tmp.substring(index, tmpLen)  
      tmp = tmp.substring(index, tmpLen)  
    }  
    if(tmp.startsWith("STP") || tmp.startsWith("ST")){ /* In case STP is part of string but was read incorretly */  
      tmp = "STP".concat(tmp.substring(3,tmp.length))  
    }  
    if(tmp.startsWith("STTP")){ /* In case STP is part of string but was read incorretly */
```

Figura 29 - Uso da biblioteca “React Native MLKit OCR”, dentro da função "getText"

Após entrar na aplicação, o utilizador pode selecionar o *icon* da câmara (presente na barra de pesquisa) e escolher uma fotografia da sua galeria ou tirar uma no momento, sendo que o resultado da leitura aparecerá de seguida, bem como os dados relativos ao veículo (Figura 30, Figura 31 e Figura 32).

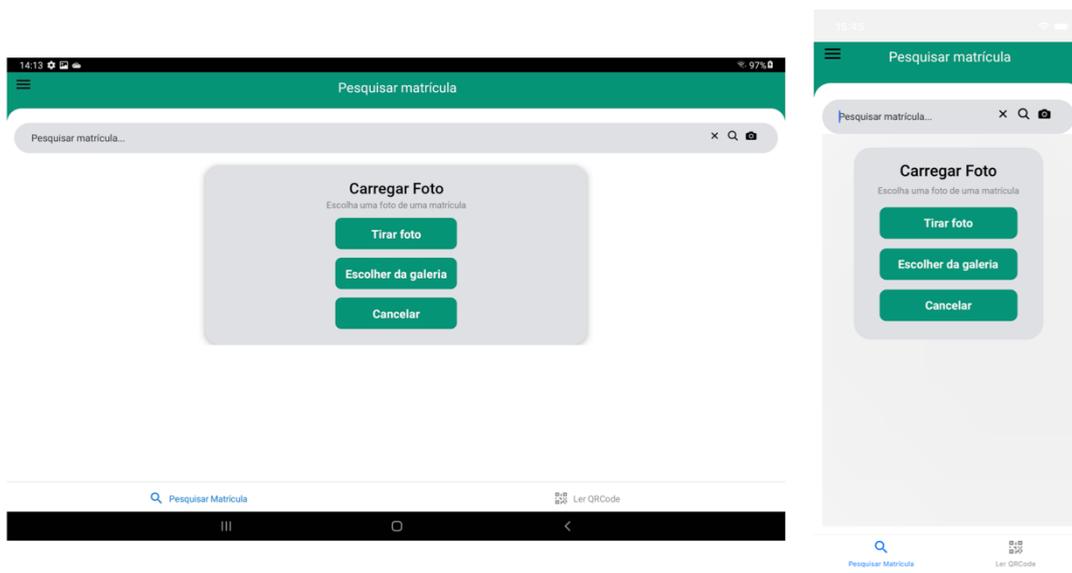


Figura 30 - Opções de pesquisa por imagem de matrícula (Android e iOS, respetivamente)

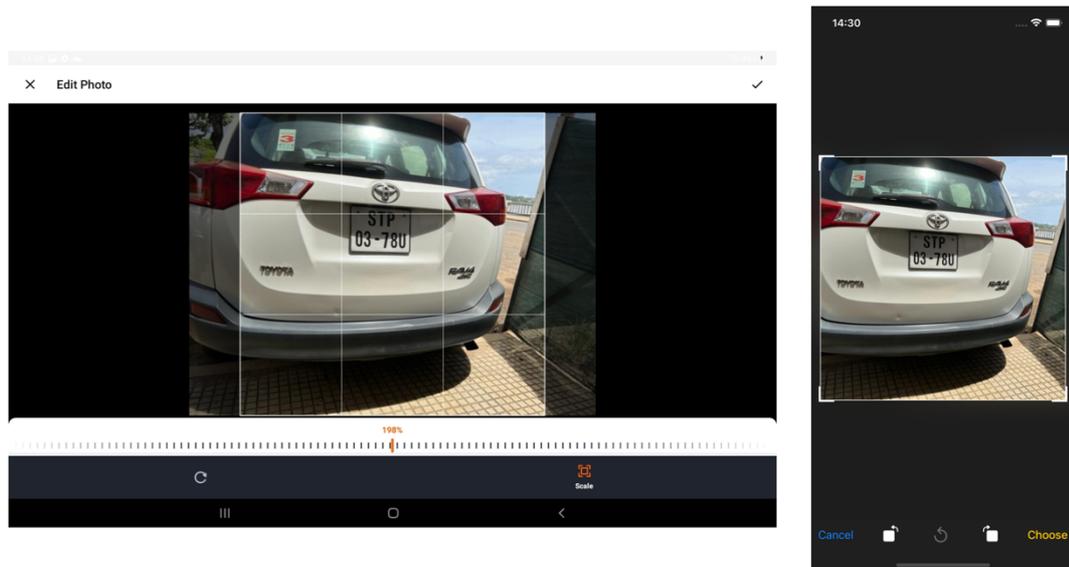


Figura 31 - Escolha de imagem e recorte da mesma (opcional)
(Android e iOS, respetivamente)

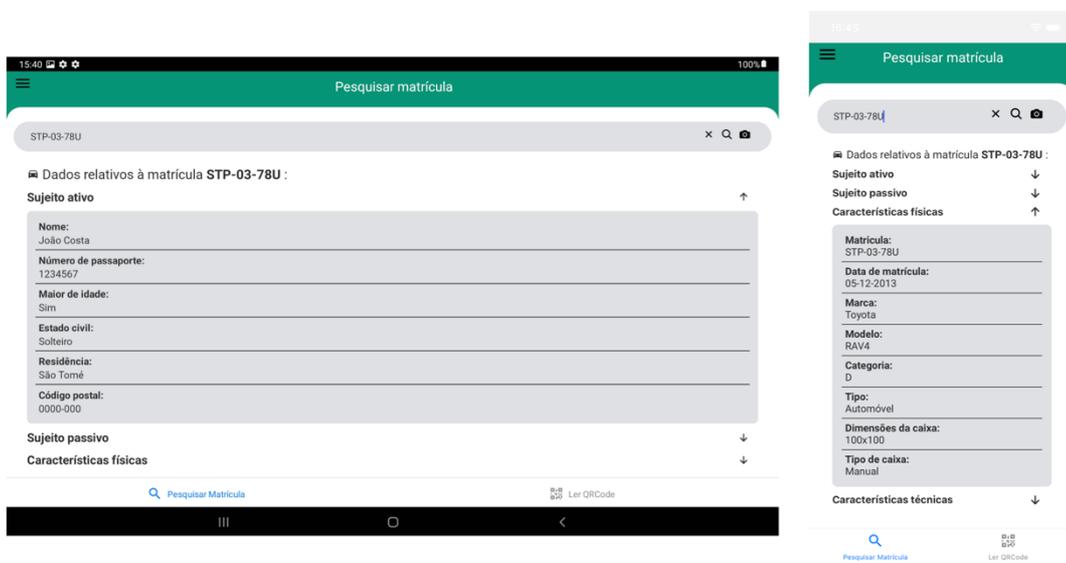


Figura 32 - Informação relativa à matrícula pesquisada (Android e iOS, respetivamente)

Leitura de texto

À medida que a aplicação foi sendo testada, com fotografias de matrículas pertencentes a veículos de São Tomé e Príncipe, foram encontradas algumas leituras de texto imprecisas. Numa primeira fase, a aplicação foi testada com matrículas em Portugal. Após isso, foi levada a São Tomé e Príncipe, onde foram realizados alguns testes informais, que permitiram a descoberta de certos erros, onde as leituras eram realizadas incorretamente (leituras estas que serão exploradas ao pormenor ainda nesta secção).

É de notar que uma matrícula, em São Tomé e Príncipe, começa por “STP”, seguindo-se de um traço e de dois caracteres (números ou letras) e,

por fim, mais um traço e dois, três ou quatro caracteres. Ou seja, uma matrícula pode ter sete, oito ou nove dígitos.



Figura 33 - Exemplo de duas matrículas, em São Tomé e Príncipe

Algumas leituras, quando incorretas, apresentavam um padrão, pelo que foram possíveis de se resolver manualmente, como o caso de ser retornado “5TP” em vez de “STP”, no início da matrícula, ou o facto das margens de algumas matrículas serem lidas como uma letra.

No entanto, existem alguns casos mais complicados de se preverem – por exemplo, por vezes, dá-se o caso de ser reconhecido um “O” em vez de um “Q”, o que também pode ser justificado com as condições de luminosidade ou estado de desgaste da matrícula. Ainda assim, o utilizador tem sempre disponível a consulta manual, sendo que pode, também, alterar o que consta de errado na cadeia de caracteres capturada, após a leitura de texto na imagem.

Caso o utilizador tente realizar a leitura de uma matrícula inválida (ou seja, que não respeita as regras acima descritas, relativamente ao formato de matrículas de São Tomé), é mostrado um aviso e a pesquisa não é efetuada (como demonstrado na Figura 34).

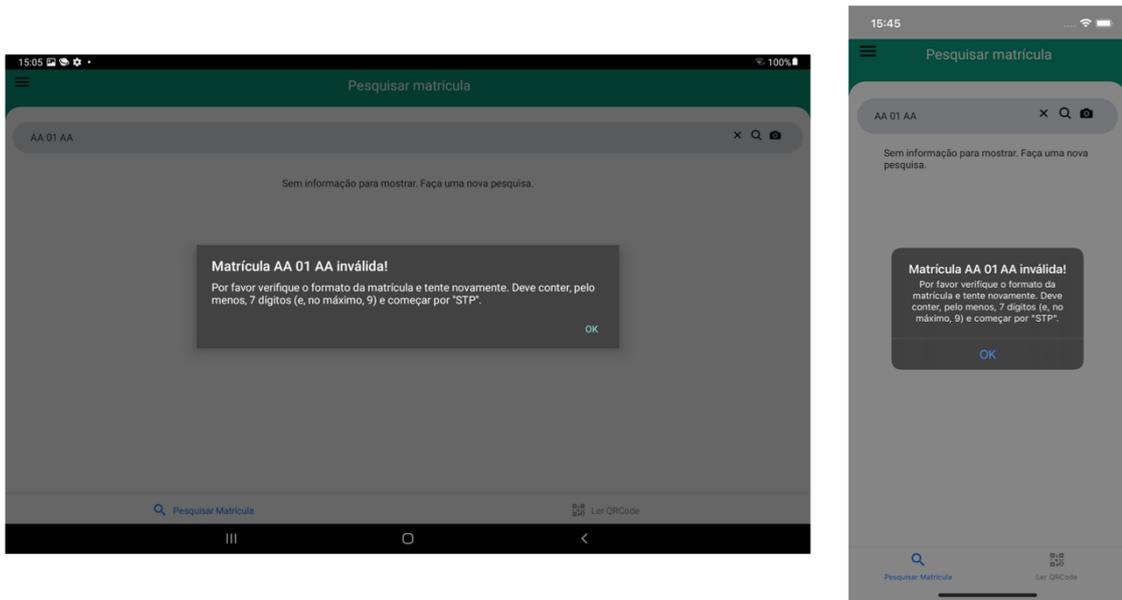


Figura 34 - Mensagem de erro - matrícula inválida (Android e iOS, respetivamente)

5.5.6 LEITURA DE QR CODE

Apesar de já terem sido garantidas as permissões de acesso à câmara, para que seja possível criar-se um leitor de *QRCode*, é necessário garantir mais uma permissão, em *Android* – a de se escrever no armazenamento externo.

```
AndroidManifest.xml X
android > app > src > main > AndroidManifest.xml
1 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
2   package="com.appautomovel">
3
4   <uses-permission android:name="android.permission.INTERNET" />
5   <uses-permission android:name="android.permission.CAMERA" />
6   <uses-feature android:name="android.hardware.camera" android:required="false" />
7   <uses-feature android:name="android.hardware.camera.front" android:required="false" />
8   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
9
10  <application
11    android:name=".MainApplication"
12    android:label="@string/app_name"
13    android:icon="@mipmap/ic_launcher"
14    android:roundIcon="@mipmap/ic_launcher_round"
15    android:allowBackup="false"
16    android:theme="@style/AppTheme">
17    <activity
18      android:name=".MainActivity"
19      android:label="@string/app_name"
20      android:configChanges="keyboard|keyboardHidden|orientation|screenLayout|screenSize|smallestScreenSize|uiMode"
21      android:launchMode="singleTask"
22      android:windowSoftInputMode="adjustResize"
23      android:exported="true">
24      <intent-filter>
25        <action android:name="android.intent.action.MAIN" />
26        <category android:name="android.intent.category.LAUNCHER" />
27      </intent-filter>
28    </activity>
29  </application>
30</manifest>
```

Figura 35 - Permissões necessárias para ser possível aceder ao armazenamento externo, em Android

Biblioteca utilizada

Para que seja possível realizar a leitura de *QR Codes*, é necessária uma biblioteca adicional – a biblioteca *React Native Camera Kit*²⁰, compatível com *iOS* e *Android*, otimizada para alta performance e com suporte para leitura de *QR Codes* e códigos de barras.

Assim, com o auxílio desta biblioteca, é possível que, após aberta a câmara (o leitor, propriamente dito), seja lido um *QR Code* e retornada a cadeia de caracteres que nele se encontra. Pretende-se que a aplicação, após deteção e leitura de um *QR Code*, feche imediatamente o leitor e que a informação que consta no *link* (*endpoint* da API) retornado pelo mesmo seja mostrada ao utilizador. Uma vez que a API ainda não foi desenvolvida, não é possível demonstrar esta funcionalidade – no entanto, pretende-se que a informação seja mostrada de acordo com a Figura 32.

Dado que o leitor de *QR Code* se apresentava deformado, caso o dispositivo estivesse na horizontal, este foi programado para ser apresentado sempre na vertical, através do uso da biblioteca *React Native Orientation Locker*. Na Figura 36, pode observar-se que, em *iOS*, visto tratar-se de um dispositivo emulador, e não de um físico, a câmara do leitor apresenta apenas uma imagem, enquanto que em *Android* é utilizada a câmara real, de um dispositivo físico.

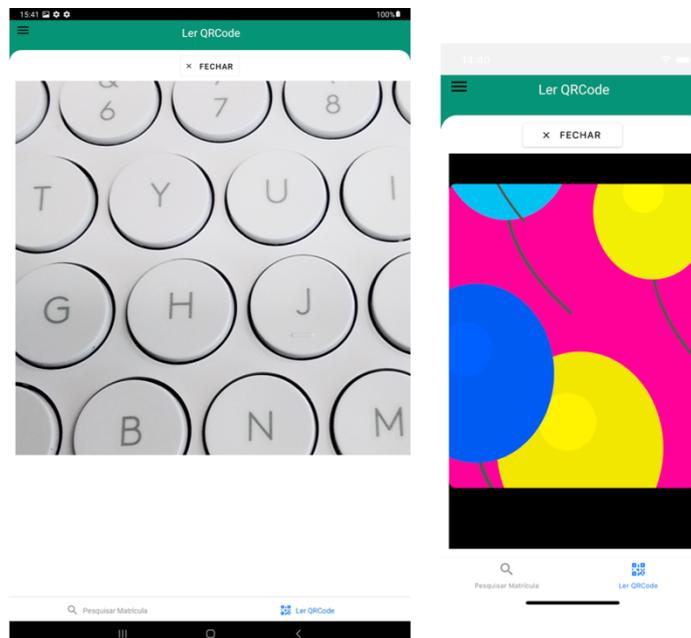


Figura 36 - Leitor de *QR Code*, através da câmara do dispositivo (*Android* e *iOS*, respetivamente)

²⁰ <https://github.com/teslamotors/react-native-camera-kit> (Acedido em 22 de julho de 2022)

5.5.7 MODO ESCURO E MODO CLARO

Uma funcionalidade extra, implementada na SIRP Auto, é a opção de se poder usar um modo escuro. Numa pesquisa rápida no *Google Trends*²¹, é possível notar-se que esta funcionalidade tem vindo ganhar uma tendência crescente, em termos de procura, nos últimos anos, visto que tem vindo a ser cada vez mais adotada. O valor “100”, nos gráficos presentes nas figuras seguintes (Figura 37 e Figura 38), representa o pico de popularidade.



Figura 37 - Tendência de pesquisas por "dark mode", no Youtube
(retirado de [59])

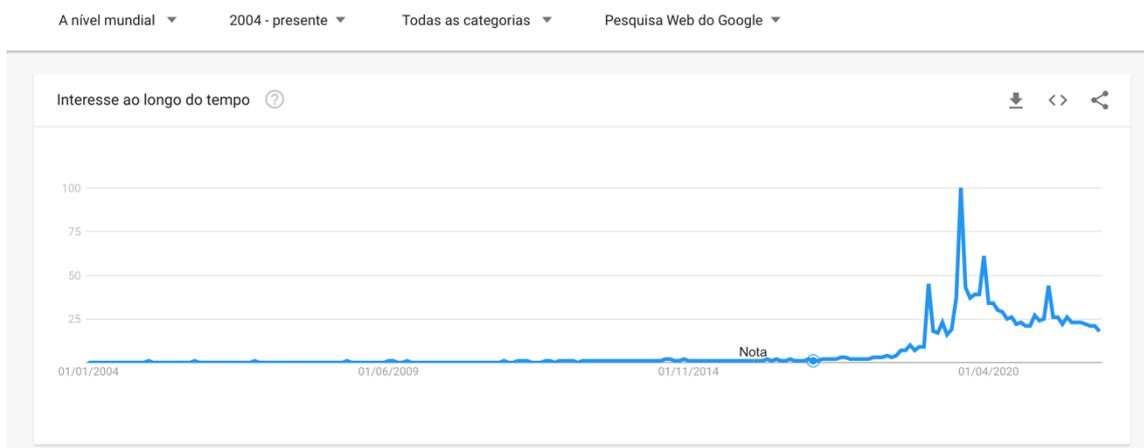


Figura 38 - Tendência de pesquisas por "dark mode", no Google
(retirado de [59])

Além disto, estudos têm vindo a demonstrar a preferência dos utilizadores – em [60], 82.7% dos inquiridos (201, no total), afirmaram ter o modo escuro ativo, nos seus dispositivos. Estes inquiridos referiram ainda que a sua escolha devia ser aplicada universalmente nos dispositivos (ou seja, as aplicações

²¹ <https://trends.google.com> (Acedido em 22 de julho de 2022)

instaladas devem respeitar essa vontade). Os que não usam este modo, justificaram-se principalmente com o facto de estarem a usar sistemas operativos onde o modo escuro ainda não estava disponível, ou porque o texto se tornava difícil de ler. [60]

Ora, os dispositivos usados para testar a aplicação, tanto com sistema operativo *Android* como com *iOS* permitem ter o modo escuro ativado no próprio dispositivo. Assim, respeitando o já mencionado anteriormente, o modo escuro foi implementado e estende-se à aplicação – caso o utilizador tenha este modo ativo no seu dispositivo, este é automaticamente ativo na aplicação, ainda que seja mostrado um aviso ao utilizador, perguntando se ele confirma esta escolha, como se observa na Figura 39.

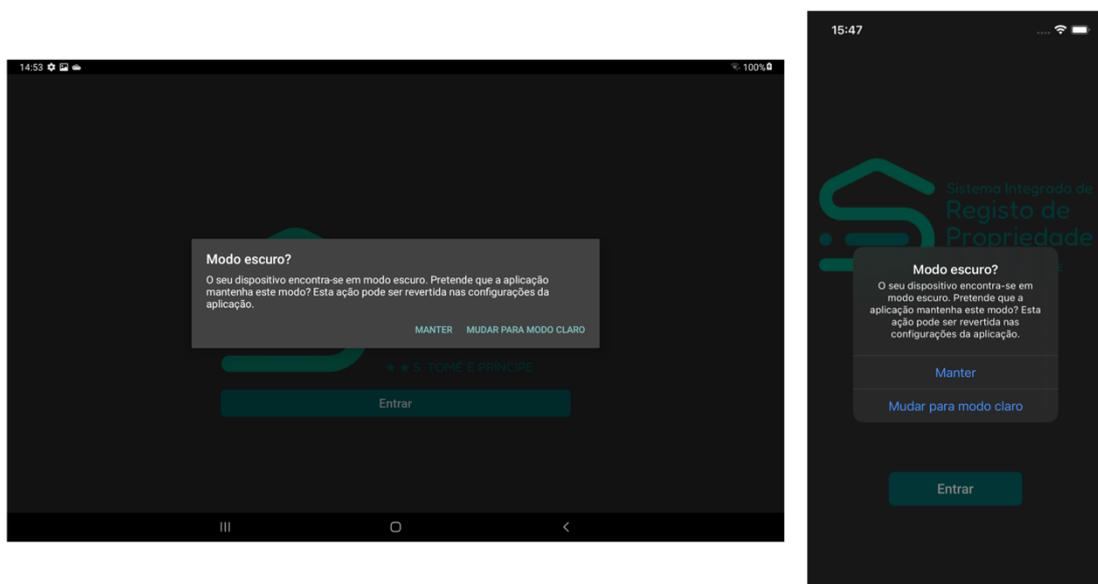


Figura 39 - Aviso inicial, relativo ao modo escuro ativo no dispositivo (*Android* e *iOS*, respetivamente)

Ainda assim, é sempre possível mudar o esquema de cores na própria aplicação, independentemente do modo que o utilizador tem ativo no seu dispositivo. Para isso, basta abrir o menu lateral esquerdo, dirigir-se à página de configurações, e escolher o modo que prefere. De forma que o novo esquema de cores fique ativo, é necessário reiniciar a aplicação, conforme é alertado ao utilizador.

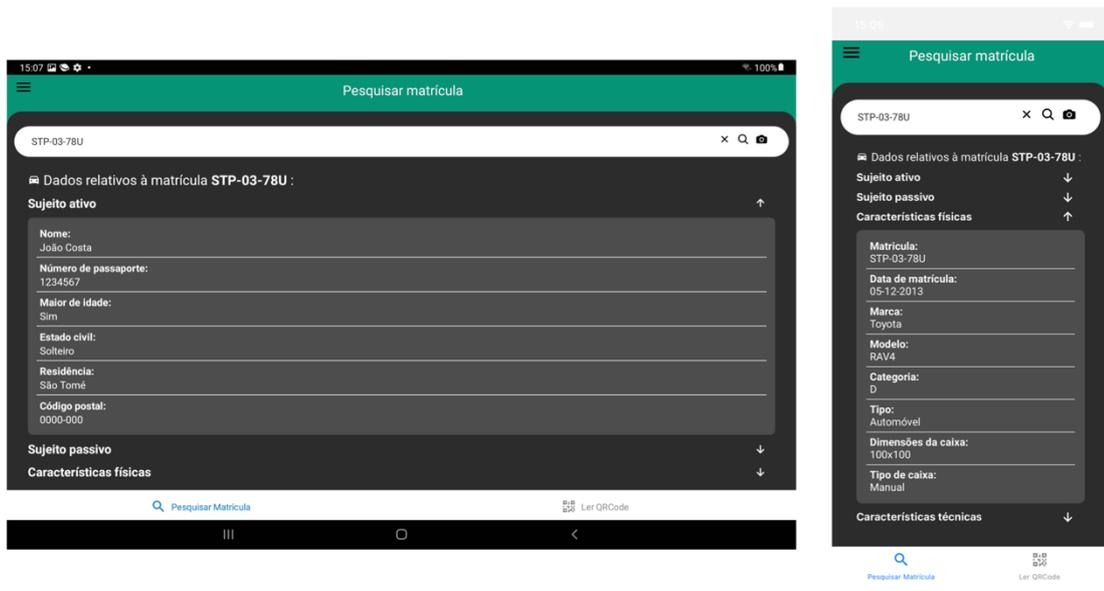


Figura 40 - Página de pesquisa em modo escuro (Android e iOS, respetivamente)

5.5.8 FREQUENTLY ASKED QUESTIONS

A página de *FAQ* (*Frequently Asked Questions*) onde constam, como o próprio nome o diz, perguntas e respostas, serve para serem esclarecidas as principais questões que possam surgir, por parte do utilizador, relativamente ao uso da aplicação. Esta página pode mesmo melhorar a experiência do utilizador ao responder às suas necessidades, aumentando também a performance da própria aplicação, ao ser usada corretamente [61].

Assim, na SIRP Auto, o utilizador tem acesso a uma página de *FAQ*, a partir do menu lateral esquerdo. Nela surgem algumas questões relativas ao uso do leitor de imagens e de *QRCode*, e também como utilizar o modo escuro.

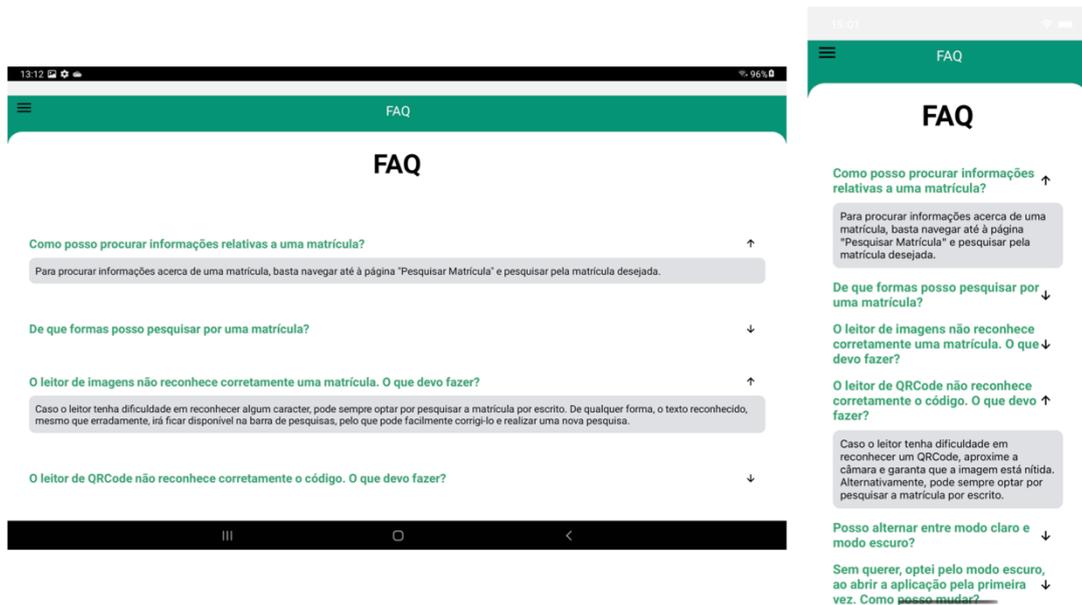


Figura 41 - Página de FAQ (Android e iOS, respetivamente)

5.5.9 GERAÇÃO DOS INSTALÁVEIS

Com o *React Native CLI*, torna-se bastante fácil gerar um ficheiro *APK* (instalável da aplicação, para *Android*). Para tal, são necessárias algumas configurações adicionais, como a criação de uma chave. Esta chave é aquela que torna possível assinar-se a aplicação, bem como gerar o ficheiro necessário para fazer o upload da aplicação para a *Google Play Store*.

Já relativamente a *iOS*, para se poder gerar o instalável da aplicação (ficheiro *.ipa*), é necessária uma licença paga de desenvolvedor da *Apple*. Como tal, não foi possível gerar o ficheiro oficial da aplicação para *iOS* – apesar de a universidade ter esta licença de desenvolvedor, iria demorar, pelo menos, seis meses até se conseguir obter o ficheiro final, devido a todas as burocracias envolvidas no processo.

6 TESTE E QUALIDADE DE SOFTWARE

No capítulo 6, é explicada a importância de testagem, quando se fala de qualidade de software. No contexto desta dissertação, foram realizados testes de integração e testes funcionais, à solução web.

6.1 CONTEXTO

Teste e qualidade de *software* é uma parte bastante importante na construção de sistemas de informação, que não deve ser descartada. Pode definir-se qualidade de *software* como sendo o grau onde o produto de *software* vai de encontro aos requisitos estabelecidos, cumprindo as expectativas do cliente e garantindo satisfação ao mesmo [62]. O *software* deve, então, procurar responder a todos os requisitos previamente acordados com o *stakeholder*, garantindo que essa resposta é dada com qualidade, eficiência e eficácia.

Desta forma, garantir a qualidade do *software* vai permitir [63]:

- Aumentar a fiabilidade, que garante que o sistema funciona sem que ocorram problemas;
- Melhorar o produto e torná-lo mais rentável;
- Aumentar e melhorar a produção;
- Baixar os custos de garantia e de *marketing*;
- Trazer satisfação aos clientes, que investiram no *software* em questão, não devem sentir que foi um desperdício de dinheiro, sendo merecedores de qualidade.

Testes de *software*

São também eles uma parte integrante da garantia de qualidade. Apesar de poder ser um processo que implica custos, uma empresa pode poupar muito em desenvolvimento, se garantir boas práticas de testagem e qualidade de *software*. Os testes permitem descobrir erros numa fase prévia da elaboração do *software*, mas também garantir fiabilidade e que está a ser construída uma aplicação de alta qualidade, sem erros. [64]

6.2 TESTES DE INTEGRAÇÃO

Os testes de integração determinam se as unidades de *software* construídas independentemente funcionam corretamente quando interligadas [65]. Com estes testes, procurou-se atingir um valor aceitável de cobertura de código, sobre os controladores existentes na solução, por forma a garantir a prevenção de erros. Atente-se que por cobertura de código entende-se a porção de código que está a ser testada pelos testes escritos.

Apesar de a tendência ser a de pensar que, quanto mais alta a percentagem de cobertura mais seguro e menos propenso a erros será o código, estudos têm mostrado o contrário. Acredita-se que a percentagem ideal de cobertura se deve encontrar entre os 70-80% - mais que isso, pode traduzir-se apenas em perda de tempo de produção, estimando-se ainda que 100% de cobertura expõe apenas cerca de metade das falhas existentes no sistema [66].

Também se encontram depoimentos, no *Google Testing Blog*²², que partilham da mesma opinião - afirmam que uma alta percentagem de cobertura não garante alta qualidade, levando a um falso sentimento de segurança. Da mesma forma, garantem que não existe uma percentagem ideal para todos os casos. Não se deve ficar obcecado a tentar atingir os 100% de cobertura - deve-se, sim, tomar passos concretos para atingir uma percentagem entre os 30-70%, assegurando sempre que o novo código vai de encontro àquilo que se pretende alcançar e que os testes focam, principalmente, no código que mais mudanças sofre. [67]

Testes aos controladores

Como referido anteriormente, os testes de integração tiveram como objetivo testar o funcionamento dos controladores. A aplicação *web* SIRP possui, atualmente, 25 controladores. Dois destes são nativos da aplicação *web* da *Microsoft* (*AccountController* e *ManageController*), estando o seu funcionamento garantido, pelo que não foram realizados testes aos mesmos. Além disso, é de notar que o controlador *Buildings* (para construir imagens) não se encontra completamente implementado, o controlador *Search* (para pesquisas na solução) foi principalmente testado por testes funcionais e o controlador *Certificates* não foi possível de testar com testes de integração, devido a algumas funcionalidades relacionadas com *CSV*. Assim sendo, foram realizados 317 testes de integração, no total. Como se pode verificar através da Figura 42, estes testes, na sua totalidade, demoram 1.1 minutos para executar.

²² <https://testing.googleblog.com> (Acedido em 22 de julho de 2022)

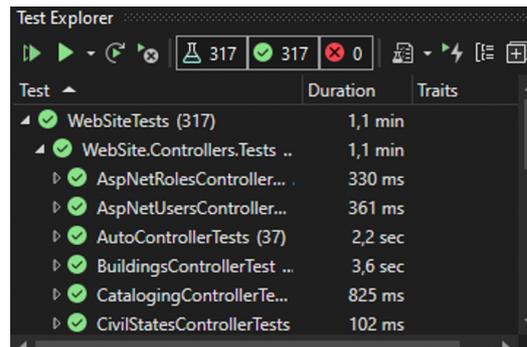


Figura 42 - Testes de integração realizados sobre os controladores

É de notar que os testes foram realizados com o recurso a uma base de dados local (réplica da base de dados de produção), criada especificamente para testes, evitando-se assim a danificação acidental dos dados e/ou estrutura da base de dados original.

Em alguns casos, os testes foram feitos com recurso a *Mock* – permite criar uma versão “falsa” de certos serviços/objetos que, assim, emulam ser os verdadeiros [65]. Isto foi útil em testes que testavam funcionalidades que apenas são acessíveis tendo *login* efetuado, com uma conta que contém as permissões necessárias para realizar essa funcionalidade, como mostram as figuras Figura 43 e Figura 44. Além disto, o *Mock* também foi útil para emular a existência de certos ficheiros e diretórios (Figura 43).

```
[TestMethod, SkipInitialize]
public void CreateSuccessTestExistingImageInDirectory()
{
    // Arrange

    //emulate user
    Mock<HttpContextBase> context = new Mock<HttpContextBase>();
    Mock<HttpRequestBase> request = new Mock<HttpRequestBase>();
    GenericIdentity identity = new GenericIdentity("jsp@ua.pt");
    identity.AddClaim(new Claim(ClaimTypes.NameIdentifier, "816e395d-ba8c-4499-99f7-925016c2cc31"));
    identity.AddClaim(new Claim(ClaimTypes.Name, "tantantantan"));
    GenericPrincipal principal = new GenericPrincipal(identity, new[] { "user" });
    context.Setup(s => s.User).Returns(principal);
    context.Setup(x => x.Request).Returns(request.Object);

    //emulate files and directories
    ConfigurationManager.AppSettings["UploadDirectory"] = "/imagesTestes";
    string testes_dir = Directory.GetCurrentDirectory().Replace("bin\\Debug", "imagesTestes");
    string images_testes_dir = Directory.GetCurrentDirectory().Replace("bin\\Debug", "imagesTestes\\PG-101");
    File.WriteAllText(Path.Combine(images_testes_dir, "PG-101-200-F.txt"), "blah blah, text");
    context.Setup(s => s.Server.MapPath("/imagesTestes")).Returns(testes_dir);
    var files = new Mock<HttpFileCollectionBase>();
    files.Setup(x => x.Count).Returns(1);
    var file = new Mock<HttpPostedFileBase>();
    MemoryStream stream = new MemoryStream(Encoding.ASCII.GetBytes("Isto é um teste"));
    file.Setup(x => x.InputStream).Returns(stream);
    file.Setup(x => x.ContentLength).Returns((int)stream.Length);
    file.Setup(x => x.ContentType).Returns("text/plain");
    file.Setup(x => x.FileName).Returns("C:/foo/PG-101-200-F.txt");
    request.Setup(x => x.Files).Returns(files.Object);
    request.Setup(x => x.Files[0]).Returns(file.Object);

    HttpPostedFileBase[] httpPostedFileBases = { file.Object };
}
```

Figura 43 - Exemplo de teste onde é usado o Mock, para emular um utilizador, diretórios e ficheiros (parte 1)

```
ImagesController controller = new ImagesController();
controller.ControllerContext = new ControllerContext(
    context.Object, new RouteData(), controller);

// Act
Task<ActionResult> actionResultTask = controller.Create(httpPostedFileBases);
actionResultTask.Wait();

//Assert
var viewResult = actionResultTask.Result;
Assert.IsNotNull(viewResult);
Assert.AreEqual(controller.TempData["Success"], "Carregamento com sucesso.<br/>1 ficheiros carregados.");
```

Figura 44 - Exemplo de teste onde é usado o Mock, para emular um utilizador, diretórios e ficheiros (parte 2)

Análise da cobertura

Para uma análise mais fácil e intuitiva, foi usada a extensão *Fine Code Coverage* (disponível no ambiente de desenvolvimento *Microsoft Visual Studio* 2022), que permite, de forma automática, gerar um ficheiro *html* onde está contido um relatório sobre a quantidade de código coberto pelos testes, em cada controlador, fornecendo também outras métricas acerca dos resultados dos testes.

Como se pode observar pela Figura 75 do Anexo 6, os controladores apresentam uma taxa de cobertura que se encontra entre os 30-70%. O objetivo foi definido tendo em conta o mencionado acima, de acordo com o artigo presente

no *Google Testing Blog* [67]: obter uma cobertura total que se encontre entre 30-70%. Isto é importante visto que o código dos controladores é aquele que mais sofre mudanças, pelo que vários erros podem ser encontrados através dos testes de integração. Ainda assim, como mostra a Figura 74 do Anexo 6, foi possível cobrir-se 88.8% das linhas da solução com testes, número bastante elevado e aceitável para cobertura.

Muitos vezes surgiram casos onde não foi de todo possível alcançarem-se os 100% de cobertura. Isto deve-se ao facto de existirem condições que não se conseguem cobrir com os testes, como é o caso de certas chavetas em que o teste não passa e, portanto, diminui a percentagem de cobertura. É de notar que o *Visual Studio* permite analisar a cobertura dos testes ao pormenor, mostrando mesmo quais as linhas de código que foram ou não cobertas. Isto vai de encontro ao dito anteriormente – tentar obter os 100% de cobertura pode traduzir-se numa perda de tempo, principalmente em termos de tempo de produção, visto que se iria estar a tentar garantir algo que nunca vai acontecer, ou que não é passível de ser coberto pelos testes.

Também na Figura 74 do Anexo 6 se podem observar dois tipos de cobertura – *line coverage* (cobertura de linhas) e *branch coverage* (afere se todos os ramos do código são avaliados – por exemplo, se existir uma condição *if/else*, existem dois possíveis ramos de serem seguidos). Apesar do *line coverage* poder ser mais importante, devido à sua elevada granularidade (ao cobrir-se uma grande percentagem de linhas, assegura-se também que a maioria dos ramos são testados), não se deve desprezar completamente o *branch coverage* – podem existir condições que só com este último serão testadas [68].

```
[TestInitialize]
public void Initialize()
{
    bool skipInitialize = GetType().GetMethod(TestContext.TestName).GetCustomAttributes<SkipInitializeAttribute>().Any();
    if (!skipInitialize)
    {
        // Initialization code here
        Image testImage = new Image
        {
            Id = Guid.NewGuid(),
            Name = string.Format("testImages [{0}]{1}", TestContext.TestName, Guid.NewGuid()),
            Filename = string.Format("testImagesFilename [{0}]{1}", TestContext.TestName, Guid.NewGuid()),
            BookSource = "PG",
            BookNumber = 100,
            BookPage = "200-F",
            ToCatalog = true,
            Notes = null,
        };
        testImage.ImageLog.Add(new ImageLog
        {
            Action = "Create",
            ImageId = testImage.Id,
            UserId = "81fe395d-ba8c-4499-99f7-925016c2cc31",
            Timestamp = DateTime.Now
        });
        db.Image.Add(testImage);
        db.SaveChanges();
        ImageId = testImage.Id;
    }
}
```

Figura 45 - Método *Initialize()*, presente nos testes de integração ao controlador *Images*

```

[TestCleanup]
public void Cleanup()
{
    bool skipCleanup = GetType().GetMethod(TestContext.TestName).GetCustomAttributes<SkipCleanupAttribute>().Any();
    if (!skipCleanup)
    {
        Image testImage = db.Image.FirstOrDefault(x => x.Name.Contains("test"));
        if (testImage != null)
        {
            db.Database.ExecuteSqlCommand("DELETE FROM [ImageLog] WHERE ImageId IN(SELECT q.ImageId FROM [ImageLog] q INNER JOIN [Image] u on(u.Id = q.Imag
            db.Database.ExecuteSqlCommand("DELETE FROM [Image] WHERE Name LIKE 'testImage%'");

            string testes_dir = Directory.GetCurrentDirectory().Replace("bin\\Debug", "lalala");
            if (System.IO.Directory.Exists(testes_dir))
            {
                DeleteDirectory(testes_dir);
            }
        }
    }
}

```

Figura 46 - Método *Cleanup()*, presente nos testes de integração ao controlador *Images*

Tal como mostram as figuras Figura 45 e Figura 46, apesar de ser usada uma base de dados dedicada unicamente a propósitos de testes, são usados dois métodos que garantem que o estado da base de dados se mantém inalterado:

- *Initialize()*: método que permite criar, na base de dados, os dados necessários para a execução de testes;
- *Cleanup()*: método que permite eliminar, da base de dados, toda a informação previamente criada durante a execução dos testes, deixando então a base de dados no estado em que se encontrava inicialmente.

6.3 TESTES FUNCIONAIS

Este tipo de testes permite avaliar as funcionalidades da aplicação de *software* [63], ao verificar-se que uma certa ação desencadeia o resultado esperado. Para realização destes, foi escolhida a *framework* de testes automatizados *Selenium*, que é gratuita. É compatível com diversos *browsers* e linguagens de programação, incluindo *C#*. Permite escrever um conjunto de instruções que são executadas diretamente no *browser*, através de um *web driver*, como se alguém estivesse a usar a aplicação. Isto permite verificar se, efetivamente, uma certa ação desencadeia o resultado esperado.

Test	Duration	Tr
WebsiteSeleniumTests (...)	1,6 min	
WebsiteSeleniumTests ..	1,6 min	
AutoTests (9)	40,7 sec	
ImagesTests (3)	10,9 sec	
LoginTest (3)	7,3 sec	
RecommendationSy...	5,5 sec	
RegisterTests (3)	11,8 sec	
SearchTests (2)	7 sec	
SignUpTest (4)	10,1 sec	

Figura 47 - Testes funcionais realizados sobre a solução

Foram realizados 25 testes funcionais, sendo o foco principal garantir que as funcionalidades mais importantes da aplicação estão a funcionar da forma desejada. Como se pode verificar através da Figura 47, estes testes, na sua totalidade, demoram 1.6 minutos para executar.

Tabela 6 - Testes funcionais realizados ao sistema

Funcionalidade	Testes realizados
Iniciar sessão	<ol style="list-style-type: none"> 1. Teste que verifica se um utilizador, previamente registado na plataforma, é capaz de iniciar sessão com sucesso; 2. Teste que verifica se, caso um utilizador tente iniciar sessão com credenciais erradas, lhe é mostrada uma mensagem de erro; 3. Teste que verifica se, caso um utilizador tente realizar sessão, mas deixa algum campo em branco (<i>e-mail</i> ou palavra-passe), lhe é mostrada uma mensagem de erro.
Registar um utilizador	<ol style="list-style-type: none"> 1. Teste que verifica se um utilizador, preenchendo todos os campos de forma válida, consegue registar-se e iniciar sessão; 2. Teste que verifica se, caso o utilizador tente fazer o registo com um utilizador já criado, uma mensagem de erro surge e o registo não acontece; 3. Teste que verifica se, caso o utilizador se tente registar com uma palavra-passe que não cumpre os requisitos necessários, uma mensagem de erro surge e o registo não acontece; 4. Teste que verifica se, caso o utilizador tente fazer o registo com campos por preencher, uma mensagem de erro surge e o registo não acontece.
Consultar informação referente a imagens de livros prediais	<ol style="list-style-type: none"> 1. Teste que verifica se, ao ser selecionado um tipo de livro, apenas são mostradas, na tabela, imagens referentes a esse livro; 2. Teste que verifica se, ao ser selecionado o número de livro, apenas são mostradas, na tabela, imagens referentes a esse livro; 3. Teste que verifica se, ao ser selecionado um utilizador, na página relativa ao histórico de ações realizadas sobre imagens, apenas são mostradas ações feitas por esse mesmo utilizador.
Consultar informação referente a registos prediais	<ol style="list-style-type: none"> 1. Teste que verifica se, ao ser selecionado um utilizador, na página relativa ao histórico de ações realizadas sobre registos, apenas são mostradas ações feitas por esse mesmo utilizador; 2. Teste que verifica se, ao ser realizada uma pesquisa por um número de registo existente, na página relativa ao histórico de ações realizadas sobre registos, é retornada informação acerca do mesmo;

	<ol style="list-style-type: none"> 3. Teste que verifica se, ao ser realizada uma pesquisa por um número de registo não existente, na página relativa ao histórico de ações realizadas sobre registos, é mostrada uma mensagem de erro.
Realizar uma pesquisa	<ol style="list-style-type: none"> 1. Teste que verifica se, ao ser realizada uma pesquisa por um termo, toda a informação retornada contém esse mesmo termo; 2. Teste que verifica se, ao ser realizada uma pesquisa por um termo, é criado um histórico para essa mesma pesquisa.
Consultar o sistema de recomendação predial	<p>Teste que verifica se, quando o utilizador pesquisa por um dado livro, apenas lhe é retornada informação acerca do mesmo.</p>
Criar registos automóveis	<ol style="list-style-type: none"> 1. Teste que verifica se, quando o utilizador seleciona “Criar novo registo”, na parte de registo automóvel, são mostradas três secções, relativas aos três registos necessários; 2. Teste que verifica se, quando o utilizador cria um registo inicial, com campos que são obrigatórios e que se encontram por preencher, é mostrada uma mensagem de erro; 3. Teste que verifica se um dado campo não cumpre o seu formato, é exibida uma mensagem de erro.
Consultar informação referente a registos automóveis	<ol style="list-style-type: none"> 1. Teste que verifica se a opção “averbar” consta nos registos do tipo “inscrição” e “registo”; 2. Teste que verifica se a opção “inscrever” consta nos registos do tipo “descrição”; 3. Teste que verifica se a opção “validar registo” consta nos registos que ainda não estão validados; 4. Teste que verifica se um registo, após validado, passa para o estado de invalidação; 5. Teste que verifica se um registo, após invalidado, passa para o estado de validação; 6. Teste que verifica se um registo que já está no estado validado, mas que sofre uma edição, passa para o estado editado.

Estrutura de um teste funcional em *Selenium*

Existem dois métodos especiais, nestes testes:

- *StartBrowser()*: método de inicialização (*SetUp*) que permite inicializar, antes de se começar a execução dos testes, o navegador que vai ser controlado automaticamente;

```
[SetUp]
public void StartBrowser()
{
    driver = new ChromeDriver("C:\\webdrivers");
}
```

Figura 48 - Método *StartBrowser*

- *CloseBrowser()*: permite terminar o processo de automatização do navegador (*TearDown*) fechando-o, depois de finalizados os testes funcionais.

```
[TearDown]
public void CloseBrowser()
{
    driver.Quit();
}
```

Figura 49 - Método *CloseBrowser*

Estes dois métodos, *StartBrowser* e *CloseBrowser*, ocorrem, respetivamente, antes e depois da realização de um teste. Tome-se como exemplo o teste de verificação da existência de opção de inscrição, nos registos do tipo “descrição”.

```
[Test]
public void inscreverOptionTest()
{
    driver.Url = "http://localhost:44359/Account/Login";
    driver.Manage().Window.Size = new Size(1080, 1000);
    driver.Manage().Timeouts().ImplicitWait = TimeSpan.FromSeconds(5);
    driver.FindElement(By.CssSelector("#input_email")).SendKeys("testeselenium@ua.pt");
    driver.FindElement(By.CssSelector("#input_password")).SendKeys("Pass123#");
    driver.FindElement(By.CssSelector("#login_button")).Click();
    var wait = new WebDriverWait(driver, new TimeSpan(0, 0, 10));
    var element = wait.Until(SeleniumExtras.WaitHelpers.ExpectedConditions.ElementIsVisible(By.Id("logged_email")));
    var email = element.Text;
    Assert.AreEqual("testeselenium@ua.pt", email);

    driver.Url = "http://localhost:44359/Auto";
    IWebElement tableElement = driver.FindElement(By.TagName("tbody"));
    IList<IWebElement> rws = tableElement.FindElements(By.TagName("tr"));
    int rws_cnt = rws.Count();

    //iterar as linhas da tabela
    for (int i = 0; i < rws_cnt; i++)
    {
        IList<IWebElement> cols = rws[i].FindElements(By.TagName("td"));
        String regType = cols[2].Text;
        IList<IWebElement> buttons = rws[i].FindElements(By.TagName("a"));
        String averbamentoButton = buttons[0].GetAttribute("title");
        if (regType.ToLower().Contains("descrição"))
        {
            Assert.IsTrue(averbamentoButton.ToLower().Contains("inscr"));
        }
    }
    driver.Close();
}
```

Figura 50 - Teste funcional para verificação da existência da opção de inscrição, nos registos do tipo “descrição”

O teste “*inscreverOptionTest*” começa por verificar se o início de sessão, por parte do utilizador (sendo este um utilizador teste, no presente contexto), é efetivamente realizado. Após esta verificação ser feita com sucesso, o teste

percorre todas as linhas da tabela onde constam os registos automóveis. Sempre que encontra um registo do tipo “descrição”, faz uma asserção por forma a verificar se esse registo contém um botão para realizar uma inscrição. Por fim, o *browser* é fechado.

6.4 CONTROLO DE VERSÕES

Como já mencionado neste documento um controlo de versões, permite gerir diferentes versões de um projeto. Desta forma, os desenvolvedores envolvidos no projeto têm a possibilidade de acompanhar um histórico da solução, podendo trabalhar paralelamente na mesma, apresentando as diferentes alterações que cada um realizou, e evitando assim duplicação ou sobreposição de código.

Para controlo de versões, no projeto SIRP, foi escolhido o gestor de repositórios *GitLab*, que permite criar ilimitados repositórios gratuitamente, públicos e privados. Além de apresentar uma *interface* bastante fácil de usar, podendo-se acompanhar o histórico de *commits* (alterações no código) e configurar o ficheiro *read.me* com diferentes tipos de informação, apresenta ainda ferramentas para integração e desenvolvimento contínuos (o que permite que sejam criadas *pipelines* e que certos procedimentos sejam assim automatizados).

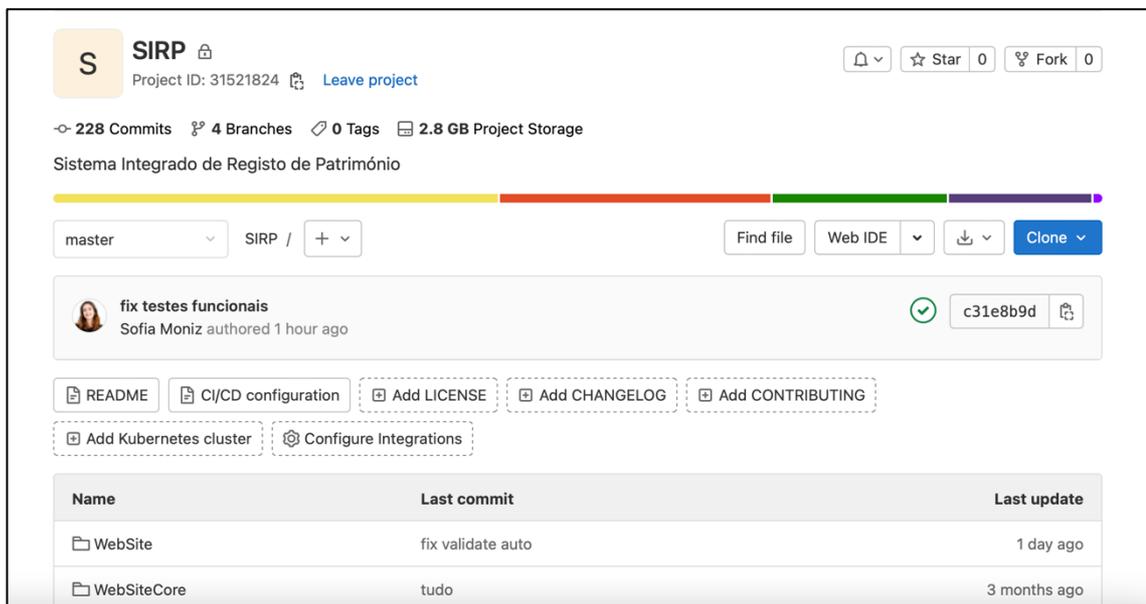


Figura 51 - Repositório "SIRP", no *GitLab*

Ao colocar-se o ficheiro *read.me* na raiz do repositório, devidamente configurado, é possível mostrar-se o seu conteúdo ao abrir o mesmo, funcionando assim como um ficheiro explicativo acerca do próprio repositório, bem como outras informações relevantes. No SIRP, o ficheiro *read.me* apresenta informação acerca do estado da solução e da estrutura da base de dados (e registo de mudanças que

7 CONCLUSÕES

No capítulo final, serão retiradas algumas conclusões acerca do desenvolvimento realizado no âmbito desta dissertação, bem como dificuldades sentidas e trabalho futuro.

Como mencionado, o projeto para construção do SIRP tem como objetivo o desenvolvimento de um sistema digital de registo de propriedade, em São Tomé e Príncipe, que reúne todas as funcionalidades necessárias para que esse registo seja efetuado de forma eficaz, eficiente e segura.

Apesar de existirem já várias soluções idênticas, mesmo em África, como estudado no segundo capítulo desta dissertação, nenhuma foi implementada neste país. A apropriação e adaptação de soluções previamente existentes foi descartada pela DGRN. Ao estudarem-se estas soluções, conseguem enumerar-se as vantagens que trouxeram aos diferentes países, onde estão incluídos países também pertencentes à África Subsariana, que já realizaram esta reforma em anos anteriores. Essas vantagens poderão vir a ser integradas na solução em desenvolvimento. Da mesma forma, a realização deste estudo mostrou que a implementação ideal para um sistema de registo de propriedades digital deve ser faseada e de melhoria contínua, o que vai de acordo ao planeado para desenvolvimento do SIRP.

A entrada em funcionamento do SIRP em São Tomé e Príncipe permitirá potenciar o crescimento económico do país, irá ainda beneficiar os cidadãos e o próprio Estado, proporcionando serviços de melhor qualidade, mais céleres, e fiáveis.

7.1 TRABALHO DESENVOLVIDO

Visto que, quando se iniciou o desenvolvimento desta dissertação, o projeto já existia, o primeiro passo dado foi a elaboração de testes de integração aos controladores da solução, dado que é também uma ótima forma para quem entra no projeto ficar a conhecer melhor o código já desenvolvido. Até então, não haviam sido feitos quaisquer tipos de testes ao sistema.

Seguidamente, foi iniciado o desenvolvimento da aplicação móvel, que partiu completamente do zero. Esta aplicação, desenvolvida em *React Native CLI*, é um passo enorme em termos de evolução do registo automóvel, visto ser a primeira fonte de pesquisa de informação para os polícias que trabalham nesta área.

Por fim, foi construída, na aplicação *web*, a parte relativa ao registo automóvel. Foram levantados todos os requisitos, de acordo com o que é preciso para registar um automóvel em São Tomé e Príncipe.

Além disto, há a destacar toda a aprendizagem que foi apreendida durante o desenvolvimento desta dissertação. A sensação de ultrapassar obstáculos que inicialmente se apresentavam intransponíveis foi vitoriosa, e o lema sempre sublinhado pelo orientador deste projeto, de que tudo deve ser feito “por partes”, foi a base de todo este processo.

7.2 DIFICULDADES SENTIDAS

Sendo um projeto já em curso, não se pode deixar de mencionar a dificuldade sentida ao integrar a dissertação num projeto que já estava em desenvolvimento, e que já contava com muito código existente. Tudo isto levou a que fosse necessário algum tempo para que se conseguisse um entendimento completo da solução em uso.

Além disso, uma vez que se trata de projeto para registo de propriedade, existem muitas definições, da área do direito, que eram até então desconhecidas. Foi precisa toda uma aprendizagem destes termos, para que fosse possível entender como devia ser a interface construída.

Não pode ainda deixar de se mencionar que ocorreram vários imprevistos, ao longo dos meses, o que levou a alguns atrasos no desenvolvimento da solução.

7.3 TRABALHO FUTURO

Existem alguns pontos que devem ser salientados, relativamente a trabalho futuro:

- Os registos, da parte automóvel, estão a ser criados com um número de registo aleatório. Isto deve ser mudado para que o número de registo provenha da base de dados – a tabela “*counter*” deve ser consultada, para se saber qual o número de registo seguinte (isto inclui, também, os registos da parte predial);
- A forma de apagar de registos, na parte automóvel, ainda se encontra por decidir – deverá ser decidido se os três registos iniciais serão eliminados no seu conjunto ou individualmente;
- O desenvolvimento da API não foi realizado, devido à falta de alguma informação, que não foi conseguida a tempo, no âmbito desta dissertação. Quando a API existir, é também necessária a sua ligação à aplicação móvel;

- A aplicação móvel deve ser alvo de mais testes, em contexto real e com o utilizador final, por forma a que seja garantida ainda mais qualidade e que se assegure que todas as funcionalidades estão corretamente implementadas e a funcionar como devido.

8 REFERÊNCIAS

- [1] E. C. Sanchez, L. Diniz, F. Meunier, and P. Rakhimova, “ Registering property,” 2013 [Online]. Available: <https://www.doingbusiness.org/content/dam/doingBusiness/media/Annual-Reports/English/DB14-Chapters/DB14-Registering-property.pdf>. [Accessed: Nov. 06, 2021]
- [2] F. Lança, “90% dos municípios sem cadastro vão avançar para regime simplificado. Meta é o mês de abril de 2021 - Justiça - Jornal de Negócios,” Oct. 20, 2020. [Online]. Available: <https://www.jornaldenegocios.pt/economia/justica/detalhe/90-dos-municipios-sem-cadastro-va-avancar-para-regime-simplificado-meta-e-o-mes-de-abril-de-2021>. [Accessed: Dec. 09, 2021]
- [3] Diário da República Eletrónico, “DECLARAÇÃO UNIVERSAL DOS DIREITOS HUMANOS.” [Online]. Available: <https://dre.pt/dre/geral/legislacao-relevante/declaracao-universal-direitos-humanos>. [Accessed: Nov. 17, 2021]
- [4] Doing Business, “Why it matters in Registering Property.” [Online]. Available: <https://www.doingbusiness.org/en/data/exploretopics/registering-property/why-matters>. [Accessed: Dec. 09, 2021]
- [5] D. McGinnis, “What Is the Fourth Industrial Revolution?,” Oct. 27, 2020. [Online]. Available: <https://www.salesforce.com/blog/what-is-the-fourth-industrial-revolution-4ir/>. [Accessed: Nov. 04, 2021]
- [6] countryeconomy, “São Tomé e Príncipe - População 2020.” [Online]. Available: <https://pt.countryeconomy.com/demografia/populacao/sao-tome-e-principe>. [Accessed: Dec. 09, 2021]
- [7] Knoema, “São Tomé e Príncipe - Densidade populacional,” 2020. [Online]. Available: <https://pt.knoema.com/atlas/São-Tomé-e-Pr%C3%ADncipe/Densidade-populacional>. [Accessed: Dec. 09, 2021]
- [8] Housing, “Digitisation of land records: Benefits for property owners and the sector.” [Online]. Available: <https://housing.com/news/digitisation-land-records-benefits-property-owners-sector/>. [Accessed: Nov. 04, 2021]
- [9] L. Diniz, F. Meunier, H. Mortada, P. Rakhimova, and J. Taras, “Registering property: The paths of digitization,” 2016 [Online]. Available: <https://www.doingbusiness.org/content/dam/doingBusiness/media/Annual-Reports/English/DB16-Chapters/DB16-CS-RP.pdf>. [Accessed: Nov. 04, 2021]

- [10] Doing Business, “Registering property,” 2011 [Online]. Available: <https://www.doingbusiness.org/content/dam/doingBusiness/media/Annual-Reports/English/DB12-Chapters/Registering-Property.pdf>. [Accessed: Nov. 04, 2021]
- [11] Lands Bureau, “Directorate of Land Registry.” [Online]. Available: <https://landsbureau.lagosstate.gov.ng/directorate-of-land-registry/>. [Accessed: Nov. 11, 2021]
- [12] D. Chukmaitova, N. de Oca, and M. Traoré, “Registering property,” 2013 [Online]. Available: <https://www.doingbusiness.org/content/dam/doingBusiness/media/Annual-Reports/English/DB13-Chapters/Registering-property.pdf>. [Accessed: Nov. 06, 2021]
- [13] Doing Business, “Registering Property - Doing Business - World Bank Group,” May 2020. [Online]. Available: <https://www.doingbusiness.org/en/data/exploretopics/registering-property>. [Accessed: Nov. 06, 2021]
- [14] Millennium Challenge Corporation, “Success Story: Early Success in Reforming Land Management in Cape Verde,” 2013 [Online]. Available: https://www.land-links.org/wp-content/uploads/2016/09/MCC_Early_Success_in_Reforming_Land_Management_in_Cape_Verde-1.pdf. [Accessed: Nov. 06, 2021]
- [15] BUPI, “Municípios.” [Online]. Available: <https://bupi.gov.pt/municipios/>. [Accessed: Nov. 09, 2021]
- [16] Singapore Land Authority, “Overview.” [Online]. Available: <https://www.sla.gov.sg/about-sla/overview>. [Accessed: Nov. 09, 2021]
- [17] Singapore Land Authority, “E-Services.” [Online]. Available: <https://www.sla.gov.sg/e-services>. [Accessed: Nov. 09, 2021]
- [18] Singapore Land Authority, “Positioning Data.” [Online]. Available: <https://www.sla.gov.sg/geospatial-development-and-services/positioning-data>. [Accessed: Nov. 09, 2021]
- [19] NSW Land Registry Services, “About Us.” [Online]. Available: <https://www.nswlrs.com.au/About-Us>. [Accessed: Nov. 10, 2021]
- [20] NSW Land Registry Services, “Land titles and access to land title information in NSW,” May 30, 2019. [Online]. Available: [https://www.nswlrs.com.au/Titles-Dealings/Title-Information-\(1\)/Access-Titling-Information](https://www.nswlrs.com.au/Titles-Dealings/Title-Information-(1)/Access-Titling-Information). [Accessed: Nov. 10, 2021]
- [21] Housing, “All about IGRS Delhi and the DORIS web portal,” Mar. 10, 2021. [Online]. Available: <https://housing.com/news/igrs-delhi/>. [Accessed: Nov. 10, 2021]
- [22] Automóvel Online, “Bem-vindo ao site AUTOMÓVEL ON-LINE! .” [Online]. Available:

- <https://www.automovelonline.mj.pt/AutoOnlineProd/>. [Accessed: Nov. 11, 2021]
- [23] F. Okeke, “Digitisation Imperatives For Nigerian Real Estate Registries,” Jun. 17, 2020. [Online]. Available: <https://www.mondaq.com/nigeria/real-estate/954908/digitisation-imperatives-for-nigerian-real-estate-registries>. [Accessed: Nov. 11, 2021]
- [24] eVRS, “FAQs.” [Online]. Available: <https://www.evrs.ng/Faqs.php>. [Accessed: Nov. 11, 2021]
- [25] eVRS, “Online Vehicle License Renewal / Registration.” [Online]. Available: <https://www.evrs.ng/index.php>. [Accessed: Nov. 11, 2021]
- [26] CGD, “Ter o Novo Documento Único Automóvel no smartphone?” [Online]. Available: <https://www.cgd.pt/Site/SaldoPositivo/mobilidade/Pages/documento-unico-automovel0507-6776.aspx>. [Accessed: Apr. 21, 2022]
- [27] C. Coronel and S. Morris, *Database systems: design, implementation & management*, 13th ed. 2018.
- [28] S. Kumar Singh, *Database Systems: Concepts, Design and Applications*, 2nd ed. 2011.
- [29] D. Sarka, M. Radivojevic, and W. Durkin, *SQL Server 2017 Developer’s Guide*. 2018.
- [30] L. Ferrari and E. Pirozzi, *Learn PostgreSQL*. 2020.
- [31] P. Raj and G. Chandra Deka, *A Deep Dive into NoSQL Databases: The Use Cases and Applications*. 2018.
- [32] Tutorialspoint, “Cassandra - Introduction.” [Online]. Available: https://www.tutorialspoint.com/cassandra/cassandra_introduction.htm. [Accessed: Jan. 05, 2022]
- [33] S. Bradshaw, E. Brazil, and K. Chodorow, *MongoDB: The Definitive Guide*, 3rd ed. 2019.
- [34] KnowledgeNile, “Understanding the Pros and Cons of MongoDB.” [Online]. Available: <https://www.knowledgenile.com/blogs/pros-and-cons-of-mongodb/>. [Accessed: Jan. 05, 2022]
- [35] J. Carpenter and E. Hewitt, Eds., *Cassandra: The Definitive Guide, (Revised)*, 3rd ed. 2022.
- [36] M. Wolfe, “MySQL vs Cassandra DB. Another RDBMS and NoSQL showdown...,” Oct. 14, 2021. [Online]. Available: <https://towardsdatascience.com/mysql-vs-cassandra-db-49bc518e1b8f>. [Accessed: Jan. 05, 2022]
- [37] Marsmedia, “Cassandra: Pros & Cons!,” Sep. 2012. [Online]. Available: <http://marsmedia.info/en/blog/cassandra-pros-cons-and-model>. [Accessed: Jan. 05, 2022]

- [38] Temok Hosting, “Difference between .NET and ASP.NET | All You Need To Know.” [Online]. Available: <https://www.temok.com/blog/difference-between-net-and-asp-net/>. [Accessed: Jan. 06, 2022]
- [39] Microsoft, “ASP.NET | Open-source web framework for .NET.” [Online]. Available: <https://dotnet.microsoft.com/en-us/apps/aspnet>. [Accessed: Jan. 06, 2022]
- [40] Microsoft, “ASP.NET MVC Pattern.” [Online]. Available: <https://dotnet.microsoft.com/en-us/apps/aspnet/mvc>. [Accessed: Jan. 06, 2022]
- [41] GeeksforGeeks, “MVC Design Pattern,” Feb. 08, 2018. [Online]. Available: <https://www.geeksforgeeks.org/mvc-design-pattern/>. [Accessed: Jan. 06, 2022]
- [42] M. Schwarzmüller, *React.js – Key Concepts*. 2022.
- [43] A. Fedosejev, *React.js Essentials*. 2015.
- [44] Nielsen Norman Group, “Mobile: Native Apps, Web Apps, and Hybrid Apps.” [Online]. Available: <https://www.nngroup.com/articles/mobile-native-apps/>. [Accessed: Jan. 10, 2022]
- [45] A. Boduch, R. Derks, and M. Sakhniuk, *React and React Native*, 4th ed. 2022.
- [46] Flutter, “Flutter - Build apps for any screen.” [Online]. Available: <https://flutter.dev/>. [Accessed: Jan. 10, 2022]
- [47] freeCodeCamp, “What is Flutter and Why You Should Learn it in 2020.” [Online]. Available: <https://www.freecodecamp.org/news/what-is-flutter-and-why-you-should-learn-it-in-2020/>. [Accessed: Jan. 10, 2022]
- [48] javatpoint, “Android Studio.” [Online]. Available: <https://www.javatpoint.com/android-studio>. [Accessed: Jan. 10, 2022]
- [49] WellDoneBy, “Android App Development: Defining Pros and Cons.” [Online]. Available: <https://welldoneby.com/blog/android-app-development-defining-pros-and-cons/>. [Accessed: Jan. 10, 2022]
- [50] J. Goodpasture, *Project Management: the Agile Way*, 2nd ed. 2016.
- [51] Agile Manifesto, “Principles behind the Agile Manifesto.” [Online]. Available: <https://agilemanifesto.org/iso/en/principles.html>. [Accessed: Jul. 22, 2022]
- [52] M. Martin, “What is a Functional Requirement in Software Engineering? Specification, Types, Examples,” Oct. 06, 2021. [Online]. Available: <https://www.guru99.com/functional-requirement-specification-example.html>. [Accessed: Nov. 18, 2021]
- [53] K. Wiegers and J. Beatty, *Software Requirements*. 2013.
- [54] S. Paradkar, *Mastering Non-Functional Requirements*. 2017.
- [55] R. Friis Dam and T. Yu Siang, “Personas – A Simple Introduction.” [Online]. Available: <https://www.interaction->

- design.org/literature/article/personas-why-and-how-you-should-use-them.
[Accessed: Jul. 22, 2022]
- [56] A. Cooper, R. Reimann, and D. Cronin, *About Face 3: The Essentials of Interaction Design*. 2007.
- [57] M. Garreta Domingo, “User Stories: As a [UX Designer] I want to [embrace Agile] so that [I can make my projects user-centered].” [Online]. Available: <https://www.interaction-design.org/literature/article/user-stories-as-a-ux-designer-i-want-to-embrace-agile-so-that-i-can-make-my-projects-user-centered>. [Accessed: Jul. 22, 2022]
- [58] Hyland, “What is Optical Character Recognition? | OCR Technology.” [Online]. Available: <https://www.hyland.com/en/resources/terminology/data-capture/what-is-optical-character-recognition-ocr>. [Accessed: Apr. 22, 2022]
- [59] “Dark Mode - Google Trends.” [Online]. Available: <https://trends.google.com/trends/explore?date=all&q=dark%20mode>. [Accessed: Apr. 27, 2022]
- [60] T. Steiner, “Let there be darkness! Maybe... Dark Mode Developer Survey Results.” [Online]. Available: <https://medium.com/dev-channel/let-there-be-darkness-maybe-9facd9c3023d>. [Accessed: Apr. 27, 2022]
- [61] House Digital, “Why are FAQs Important? | How to Write a Good FAQ Page.” [Online]. Available: <https://www.housedigital.co.uk/why-faqs-are-important-to-your-seo-strategy-and-how-to-write-a-good-faq-page/>. [Accessed: May 25, 2022]
- [62] D. Galin, *Software Quality*. 2018.
- [63] A. Sayed Mahfuz, *Software Quality Assurance*. 2016.
- [64] A. Spillner and T. Linz, *Software Testing Foundations*, 5th ed. 2021.
- [65] M. Aniche, *Effective Software Testing*. 2022.
- [66] S. Cornett, “Minimum Acceptable Code Coverage.” [Online]. Available: <https://www.bullseye.com/minimum.html>. [Accessed: Feb. 22, 2022]
- [67] C. Arguelles, M. Ivanković, and A. Bender, “Code Coverage Best Practices,” Aug. 07, 2020. [Online]. Available: <https://testing.googleblog.com/2020/08/code-coverage-best-practices.html>. [Accessed: Feb. 22, 2022]
- [68] Ardalis, “Which is more important, line coverage or branch coverage?,” Aug. 27, 2019. [Online]. Available: <https://ardalis.com/which-is-more-important-line-coverage-or-branch-coverage/>. [Accessed: May 26, 2022]

ANEXO 1 – PERSONAS

No Anexo 1 serão descritas, com detalhe, as personas do sistema.

Tabela 7 - Personas existentes no sistema

Paula Borges	Casada, 53 anos, residente em São Tomé e Príncipe. Funcionária do registo predial há mais de 20 anos, faz parte do seu dia a dia a atualização de registos – descrever propriedades, inscrever-lhes e/ou averbar-lhes factos e emitir certidões. Sente necessidade da existência de uma ferramenta de pesquisa, visto que depende muito do seu tempo de trabalho a procurar informação nos diversos livros existentes.
Sílvia Rocha	Casada, 48 anos, residente em São Tomé e Príncipe. Como diretora do registo predial, é responsável pelo pagamento de emolumentos, se aplicável, aos funcionários. Assim, necessita de conhecer, com exatidão, as estatísticas relativas ao desempenho de cada um.
Roberto Santos	Solteiro, 22 anos, residente em São Tomé e Príncipe e estudante de engenharia informática. Para ganhar algum dinheiro, opta por fazer alguns trabalhos extra nos seus tempos livres.
Nelson Viegas	Casado, 45 anos, residente em São Tomé e Príncipe, confirma os registos de propriedade que são catalogados, garantindo assim que não existem erros na inserção dos mesmos.
Cíntia Boa Morte	Casada, 42 anos, residente em São Tomé e Príncipe, é conservadora do registo de propriedade do país há mais de 15 anos. Assim, tem total conhecimento sobre quais as informações/campos que são necessários aquando do processo de registo, bem como a obrigatoriedade desses campos.
Raimundo Silvério	Casado, 58 anos, residente em São Tomé e Príncipe, tem alto conhecimento sobre leis e burocracias associadas ao registo de propriedade no país. Desta forma, é capaz de gerir registos, na medida em que consegue encontrar irregularidades, bloqueando-os. No entanto, sente que o processo ainda precisa de evoluir, pois o processo de fazer toda esta verificação em papel é bastante dispendiosa em termos de tempo.
Almerindo D'Abreu	Casado, 53 anos, residente em São Tomé e Príncipe e polícia, enfrenta dificuldades no seu trabalho, quando realiza operações de fiscalização automóvel e verificação de documentos relacionados, visto não conseguir, de forma rápida e fácil, obter informações relativas a veículos. Não tem nenhuma forma para realizar esta pesquisa, caso não tenha acesso ao documento único do veículo, tornando-se um processo demorado, ou até mesmo impossível. Sente que, caso conseguisse realizar pesquisas sobre os mesmos e receber, em retorno, informações atualizadas, o seu trabalho seria muito mais útil e eficiente.

ANEXO 2 – DIAGRAMAS DE SEQUÊNCIA (REGISTO AUTOMÓVEL)

No segundo anexo deste documento serão descritas, com detalhe, as ações necessárias para realizar cada operação relacionada com o registo automóvel.

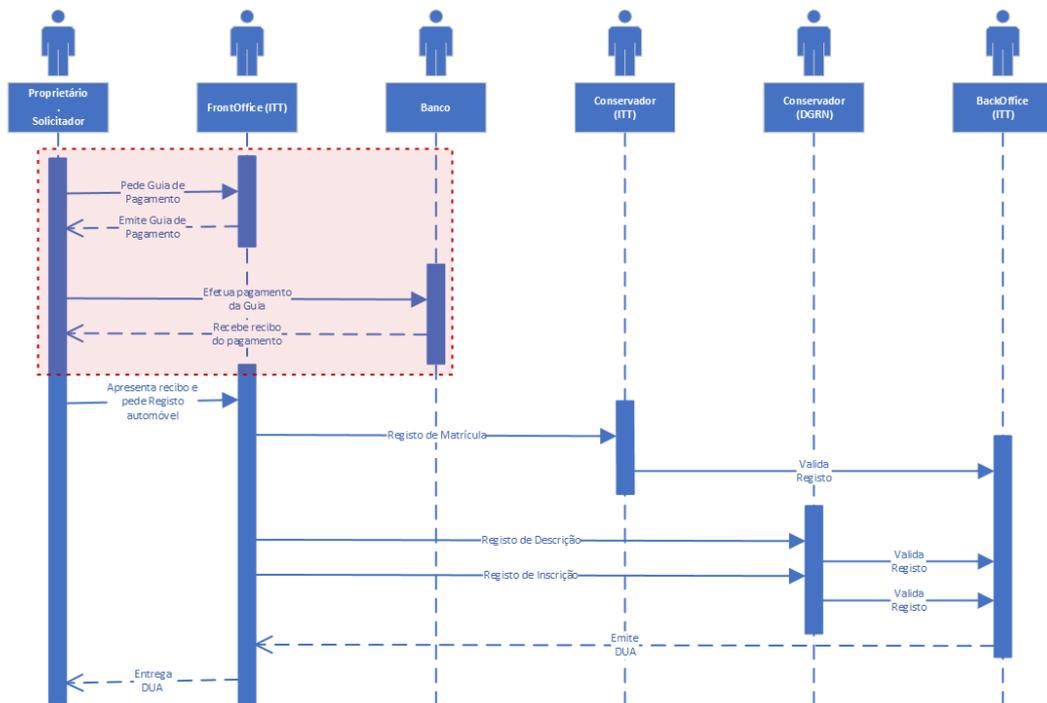


Figura 54 - Diagrama de sequência relativo ao registo inicial e emissão de DUA

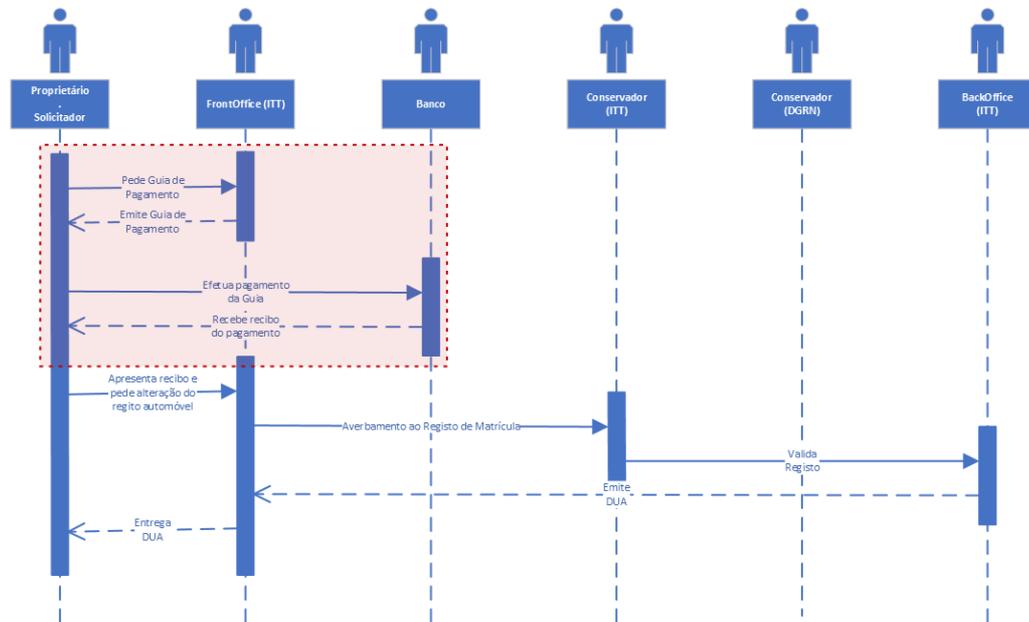


Figura 55 - Diagrama de sequência relativo à alteração das características físicas de um veículo

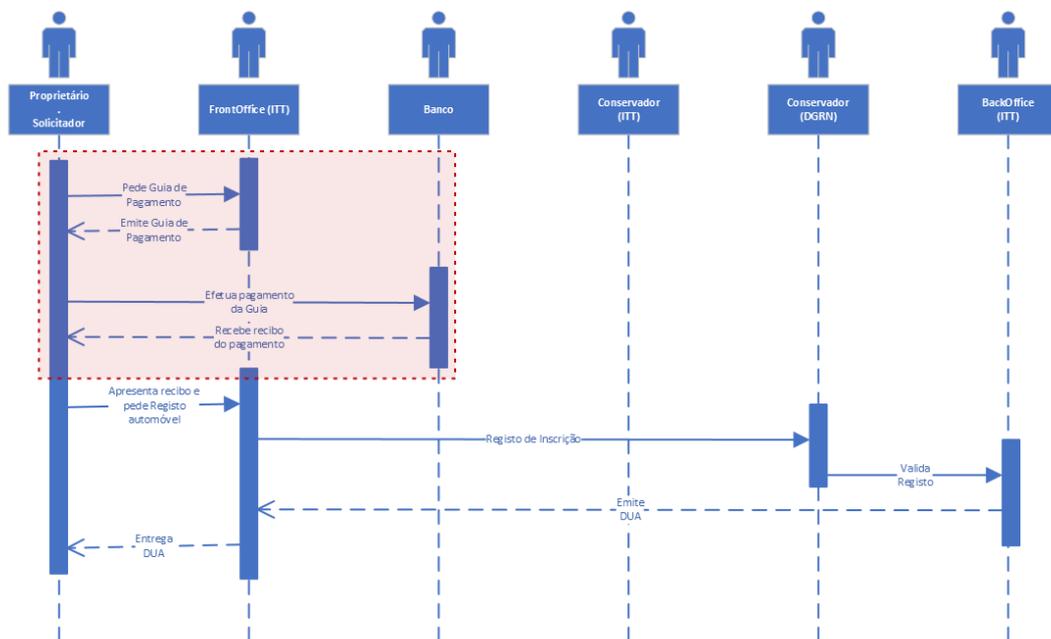


Figura 56 - Diagrama de sequência relativo à transmissão de um veículo

ANEXO 3 – ESTRUTURA DA BASE DE DADOS

No Anexo 3 deste documento será descrita, com detalhe, a estrutura da base de dados do sistema.

Tabela 8 - Descrição das tabelas criadas na base de dados

Register	Tabela que armazena todos os registos de propriedade realizados no SIRP.
RegisterLog	Tabela que armazena o histórico de registos de propriedade realizados no SIRP, por modo a poderem controlar-se os registos feitos por cada utilizador, e qual a ação efetuada nesse registo.
Template	Tabela que armazena as máscaras de catalogação (<i>templates</i>) dos diferentes registos que podem ser feitos, no SIRP. Cada <i>template</i> tem um conjunto de regras a seguir associadas.
TemplateType	Tabela que armazena os tipos de <i>templates</i> existentes.
TemplateField	Tabela que armazena os campos (<i>fields</i>) presentes em cada <i>template</i> .
Field	Tabela que armazena os campos (<i>fields</i>) que pertencem aos <i>templates</i> e serão preenchidos nos registos.
FieldType	Tabela que associa um identificador a cada tipo de campo.
Triplet	Tabela que armazena os valores referentes a cada um dos campos (<i>fields</i>) de um registo.
Person	Tabela que armazena a informação relativa às pessoas (singular ou empresa) que constam em cada registo.
CivilState	Tabela que armazena o estado civil de cada pessoa que consta em cada registo.
Image	Tabela que armazena todas as imagens de livros, digitalizadas, e detalhes acerca das mesmas. Estas têm um formato específico. Por exemplo, na imagem “PG-027-001-F”, “PG” indica o livro fonte ao qual a imagem pertence, “027” indica o número desse mesmo livro, “001” indica o número da folha, e “F” indica que é a “frente” da folha (este campo pode também tomar o valor “V”, caso se trate do verso”).
ImageLog	Tabela que armazena o histórico de ações realizadas sobre imagens, no SIRP, por modo a poderem controlar-se as ações feitas sobre uma imagem, bem como qual o utilizador que realizou a ação.
SearchLog	Tabela que armazena o histórico de pesquisas realizadas, no SIRP, por modo a ter-se controlo sobre o que é pesquisado por cada utilizador.

<i>Counter</i>	Tabela que serve como um contador de páginas, páginas estas que formarão novos livros de registo de propriedade, a partir dos registos criados digitalmente.
<i>UsersActivityLog</i>	Tabela que armazena o histórico de ações realizadas por cada utilizador.

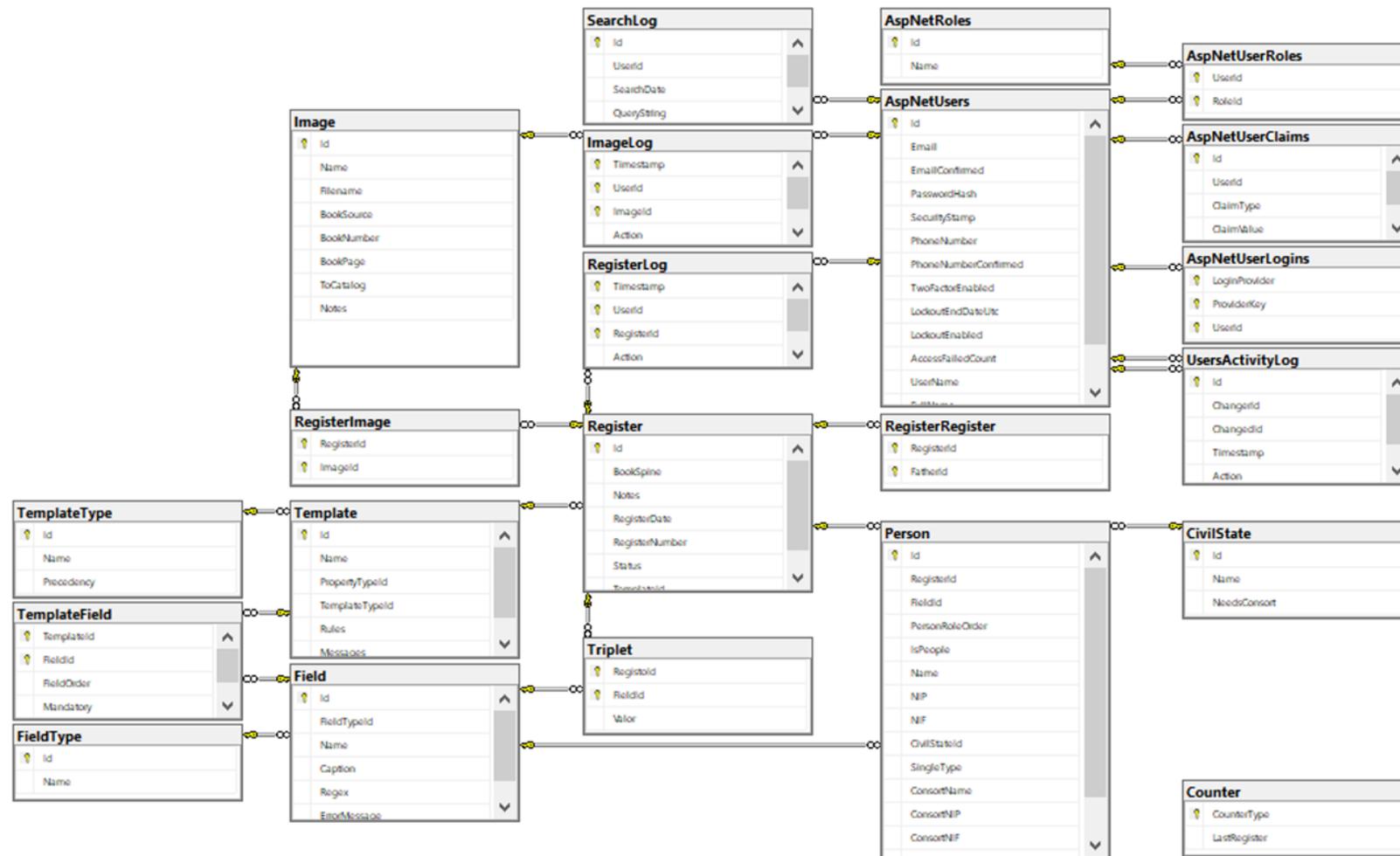


Figura 57 - Diagrama da base de dados do sistema

ANEXO 4 – REGISTO PREDIAL (APLICAÇÃO WEB)

No Anexo 4 serão mostradas algumas páginas *web*, relativas ao registo predial, por forma a dar algum contexto relativamente ao que já se encontrava implementado neste projeto.

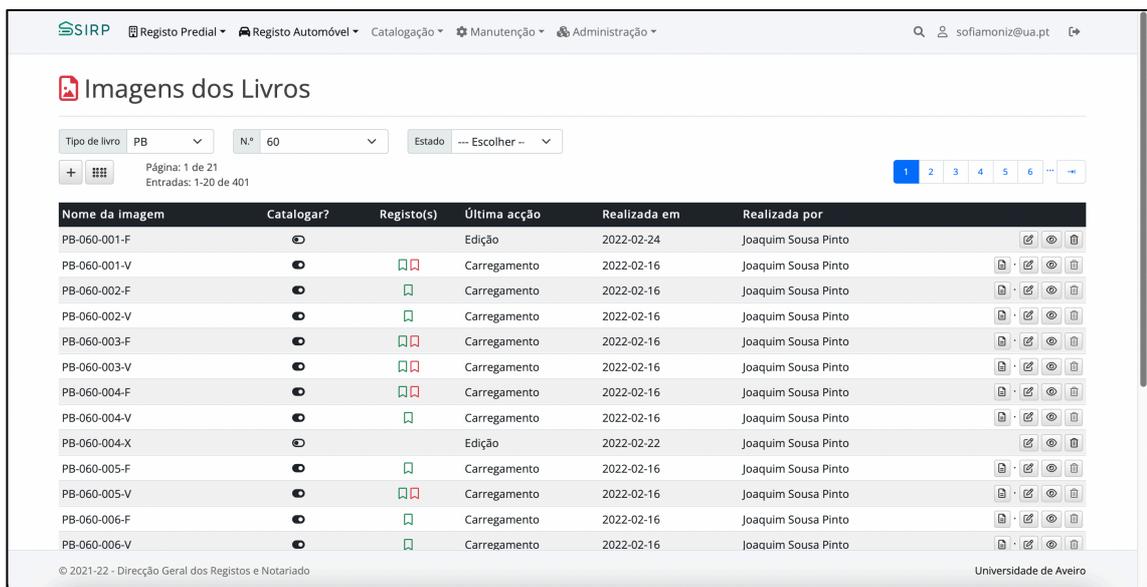


Figura 58 - Página para consulta de imagem dos livros

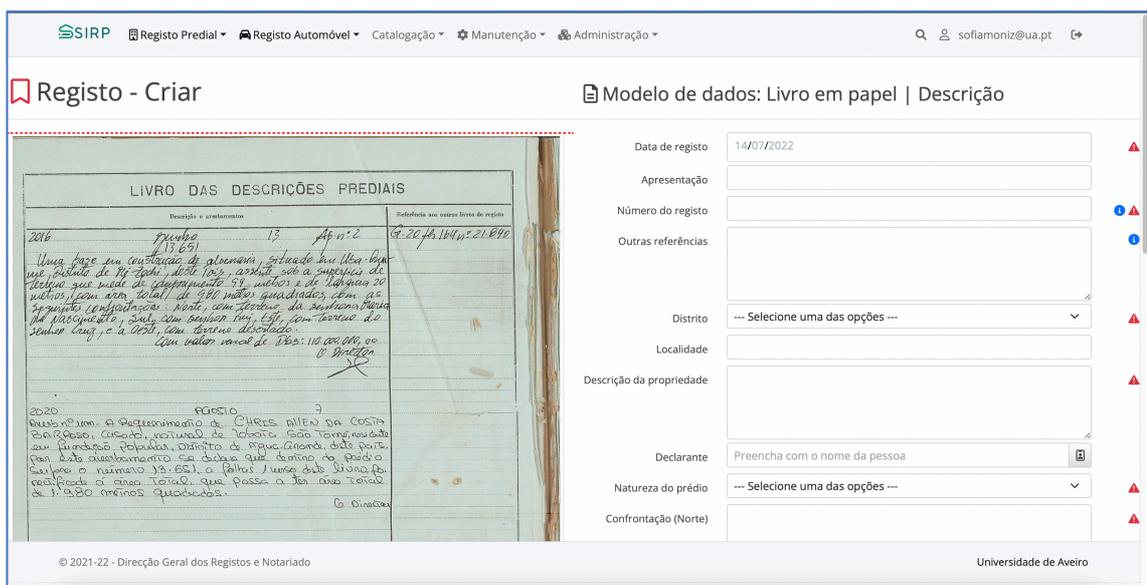


Figura 59 - Página para catalogação de uma imagem

SIRP Registo Predial Registo Automóvel Catalogação Manutenção Administração

sofiamoniz@ua.pt

📖 Livro de Registos [PB-060]

Insira um número de registo

Tipo de livro: PB N.º: 60 Estado: --- Escolher -

Página: 1 de 21 Entradas: 1-20 de 405

N.º Registo	Folha(s)	Tipo de registo	Estado	Última acção	Realizada em	Realizada por	
13651	001-V	Livro em papel Descrição	●	Validação	2022-05-04	Ilma Salvaterra	A
13651/1	001-V	Livro em papel Averbamento à Descrição	●	Validação	2022-05-04	Ilma Salvaterra	
13652	002-F	Livro em papel Descrição	●	Validação	2022-07-01	Ilma Salvaterra	A
13653	002-V	Livro em papel Descrição	●	Validação	2022-07-01	Ilma Salvaterra	A
13654	003-F	Livro em papel Descrição	○	Catalogação	2022-02-22	Joaquim Sousa Pinto	A
13654/1	003-F	Livro em papel Averbamento à Descrição	○	Catalogação	2022-02-27	Joaquim Sousa Pinto	
13655	003-V	Livro em papel Descrição	○	Edição	2022-02-22	Joaquim Sousa Pinto	A
13655/1	003-V	Livro em papel Averbamento à Descrição	○	Catalogação	2022-03-19	Joaquim Sousa Pinto	
13656	004-F	Livro em papel Descrição	○	Edição	2022-02-22	Elissandro Kelman Torres	A
13656/1	004-F	Livro em papel Averbamento à Descrição	○	Catalogação	2022-03-02	Joaquim Sousa Pinto	
13657	004-V	Livro em papel Descrição	○	Catalogação	2022-02-22	Joaquim Sousa Pinto	A
13658	005-F	Livro em papel Descrição	●	Validação	2022-07-01	Ilma Salvaterra	A

© 2021-22 - Direcção Geral dos Registos e Notariado Universidade de Aveiro

Figura 60 - Página para consulta dos livros de registo

ANEXO 5 – REGISTO AUTOMÓVEL (APLICAÇÃO WEB)

No presente anexo serão mostradas algumas páginas *web*, relativas ao registo automóvel.

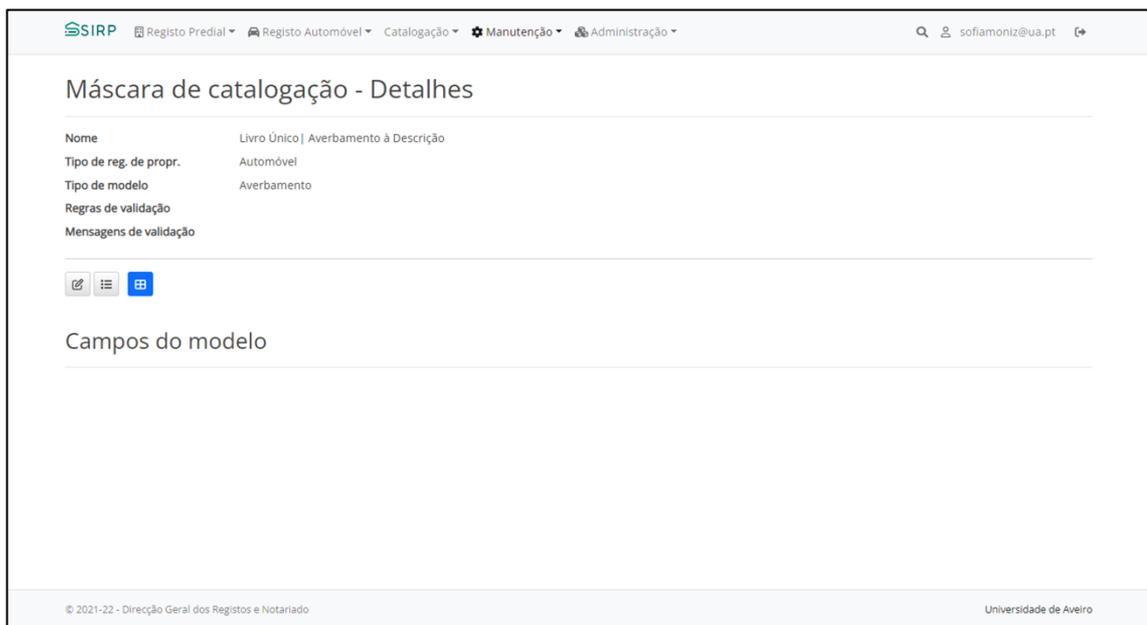


Figura 61 - Página de detalhes de uma máscara de catalogação

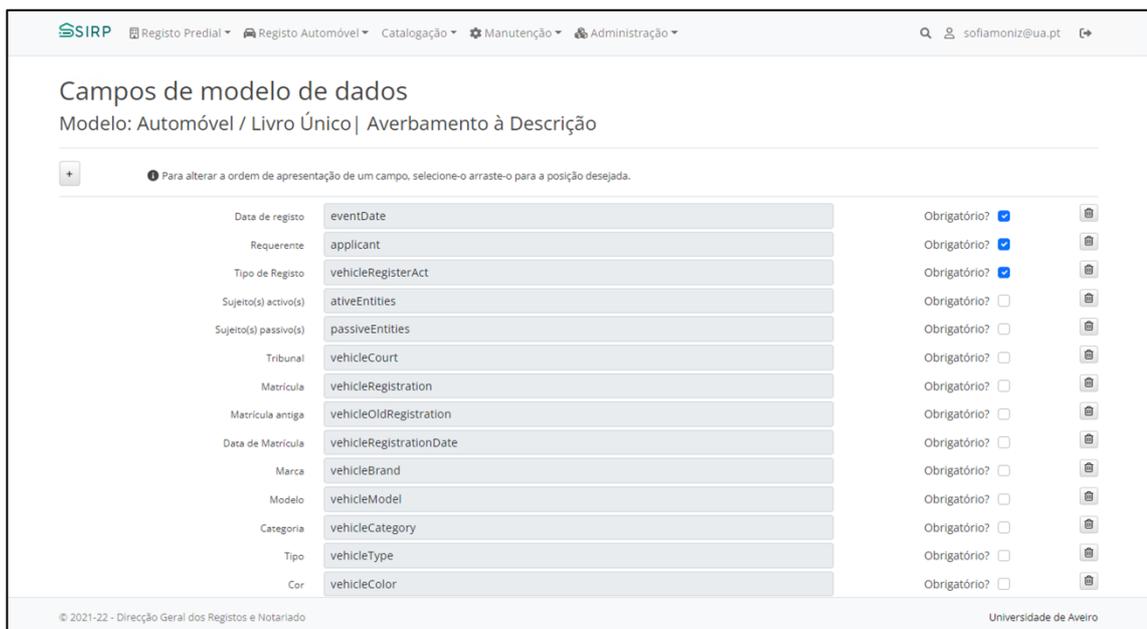


Figura 62 - Página relativa aos campos de uma máscara de catalogação

The screenshot shows the 'Registo Automóvel' form in the SIRP system. The form is divided into several sections:

- Data de registo:** A date input field with a placeholder 'mm/dd/yyyy' and a calendar icon.
- Sujeito(s) activo(s):** A dropdown menu followed by a text input field with a placeholder 'Preencha com o nome da pessoa' and a trash icon.
- Sujeito(s) passivo(s):** A dropdown menu followed by a text input field with a placeholder 'Preencha com o nome da pessoa' and a trash icon.
- Quota-Parte:** A text input field.
- Matrícula:** A text input field.
- VIN:** A text input field.
- Marca:** A text input field.
- Modelo:** A text input field.
- Notas de catalogação:** A large text area for additional information.

 The top navigation bar includes 'SIRP', 'Registo Predial', 'Registo Automóvel', 'Catalogação', 'Manutenção', and 'Administração'. The user's email 'sofiamoniz@ua.pt' is visible in the top right. The footer contains '© 2021-22 - Direcção Geral dos Registos e Notariado' and 'Universidade de Aveiro'.

Figura 63 – Página para criação de um registo automóvel inicial

The screenshot shows the 'Livro de Registos [AU-001]' page in the SIRP system. The page features a search bar 'Insira um número de registo' and filters for 'Tipo de livro' (AU) and 'Estado' (Escolher). Below the filters, it indicates 'Página: 1 de 1' and 'Entradas: 1-6 de 6'. The main content is a table of records:

N.º Registo	Folha(s)	Tipo de registo	Estado	Última acção	Realizada em	Realizada por	
1		Livro Único Inscrição	○	Catalogação	Hoje, 13:10:31	Sofia Moniz	A ✓ ↻ 🗑️ 📄
114		Livro Único Inscrição	○	Catalogação	Hoje, 13:07:25	Sofia Moniz	A ✓ ↻ 🗑️ 📄
114 / 1		Livro Único Averbamento à Inscrição	○	Catalogação	Hoje, 13:08:21	Sofia Moniz	✓ ↻ 🗑️ 📄
255		Livro Único Registo	○	Catalogação	Hoje, 13:07:25	Sofia Moniz	A ✓ ↻ 🗑️ 📄
255 / 1		Livro Único Averbamento ao Registo	○	Catalogação	Hoje, 13:09:51	Sofia Moniz	✓ ↻ 🗑️ 📄
271		Livro Único Descrição	○	Catalogação	Hoje, 13:07:25	Sofia Moniz	I ✓ ↻ 🗑️ 📄

Below the table, a legend explains the symbols:

- - Registo catalogado.
- - Registo editado.
- - Registo validado.
- - Registo bloqueado.
- - Registo inválido.
- 🗑️ - Notas.

 A loading indicator '75 ms' is shown at the bottom left. The footer contains '© 2021-22 - Direcção Geral dos Registos e Notariado' and 'Universidade de Aveiro'.

Figura 64 - Página para consulta do livro de registos

The screenshot shows the 'Registo - Editar' page for an initial vehicle registration. The page is titled 'Registo - Editar' and has a navigation bar with 'Registo Predial', 'Registo Automóvel', 'Catalogação', 'Manutenção', and 'Administração'. The user is logged in as 'sofiamoniz@ua.pt'. The page is divided into three tabs: 'Descrição', 'Inscrição', and 'ITT'. The 'Inscrição' tab is active. The form contains the following fields:

- Data de registo: 05/24/2022
- Sujeito(s) activo(s): José Fernandes
- Sujeito(s) passivo(s): António Almeida
- Quota-Parte: 100%
- Matricula: STP-03-78U
- VIN: 26536475894875647
- Marca: Toyota
- Modelo: Yaris
- Notas de catalogação: (empty text area)

At the bottom, there is a footer with '© 2021-22 - Direcção Geral dos Registos e Notariado' and 'Universidade de Aveiro'.

Figura 65 - Página de edição de um registo automóvel inicial

The screenshot shows the 'Registo - Editar' page for an 'Averbamento à Inscrição' (singular registration). The page is titled 'Registo - Editar' and has a navigation bar with 'Registo Predial', 'Registo Automóvel', 'Catalogação', 'Manutenção', and 'Administração'. The user is logged in as 'sofiamoniz@ua.pt'. The page is divided into three tabs: 'Descrição', 'Inscrição', and 'ITT'. The 'Inscrição' tab is active. The form contains the following fields:

- Data de registo: 05/24/2022
- Requerente: António Almeida
- Nome/Firma anterior: José Fernandes
- Sujeito(s) activo(s): Preencha com o nome da pessoa
- Sujeito(s) passivo(s): Preencha com o nome da pessoa
- Menções especiais: (empty text area)
- Notas de catalogação: (empty text area)

At the bottom, there is a footer with '© 2021-22 - Direcção Geral dos Registos e Notariado' and 'Universidade de Aveiro'.

Figura 66 - Página de edição de um averbamento (registo "singular")

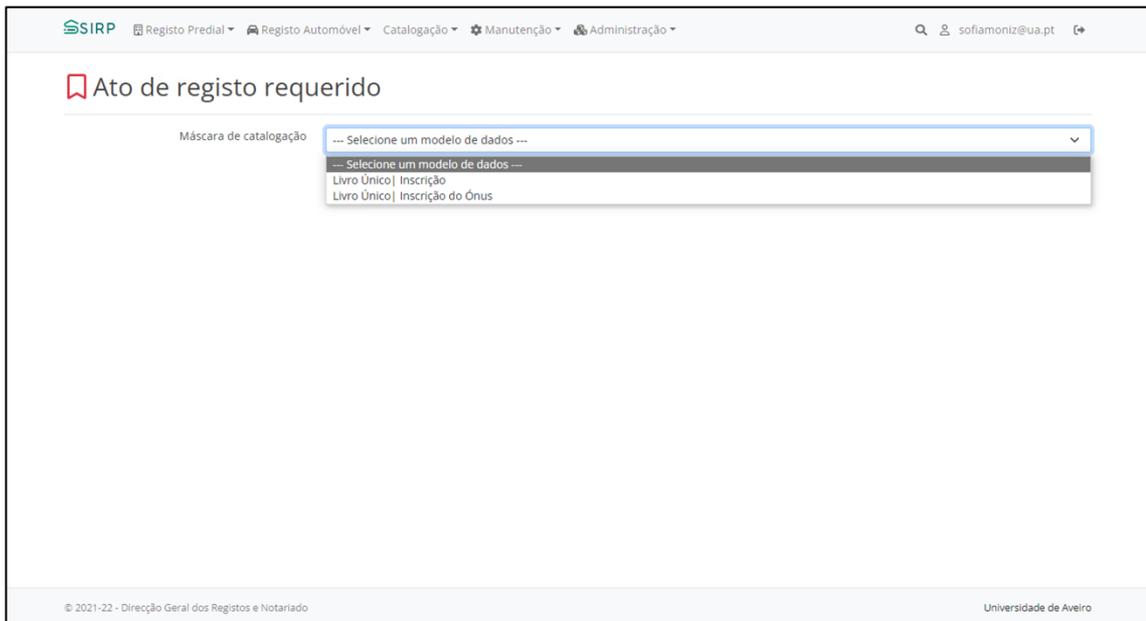


Figura 67 - Página para escolha do tipo de inscrição a realizar

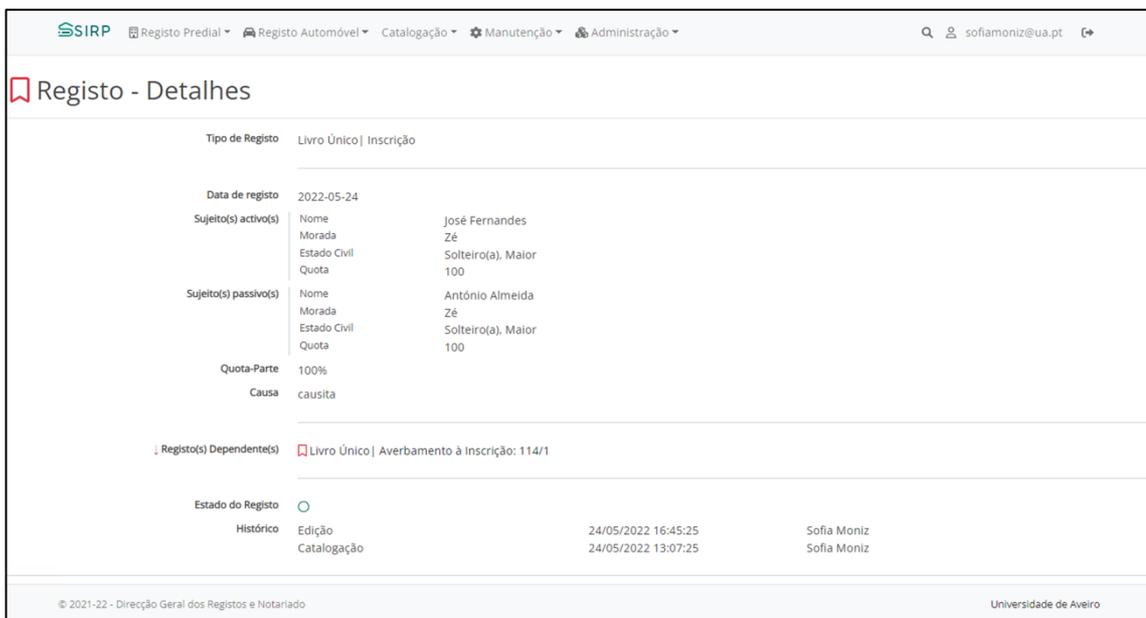


Figura 68 - Página de visualização de detalhes de um registo

The screenshot shows the 'Ato de registo requerido' page. At the top, there is a navigation bar with 'SIRS' and menu items: 'Registo Predial', 'Registo Automóvel', 'Catalogação', 'Manutenção', and 'Administração'. A search bar contains 'sofiamoniz@ua.pt'. The main heading is 'Ato de registo requerido'. Below it, there is a label 'Máscara de catalogação' and a dropdown menu. The dropdown menu is open, showing the following options: '--- Selecione um modelo de dados ---', '--- Selecione um modelo de dados ---', 'Livro Único | Averbamento à Descrição', 'Livro Único | Averbamento à Inscrição' (highlighted), 'Livro Único | Averbamento à Inscrição do Ônus', and 'Livro Único | Averbamento ao Registo'. At the bottom, there is a footer with '© 2021-22 - Direcção Geral dos Registos e Notariado' and 'Universidade de Aveiro'.

Figura 69 - Página para escolha do tipo de averbamento a realizar

The screenshot shows the 'Registo - Validar' page. At the top, there is a navigation bar with 'SIRS' and menu items: 'Registo Predial', 'Registo Automóvel', 'Catalogação', 'Manutenção', and 'Administração'. A search bar contains 'sofiamoniz@ua.pt'. The main heading is 'Registo - Validar'. Below it, there is a label 'Modelo de dados: Livro Único | Registo'. The form contains the following fields: 'Data de registo' (05/24/2022), 'Sujeito(s) activo(s)' (José Fernandes), 'Sujeito(s) passivo(s)' (António Almeida), 'Matrícula' (STP-03-78U), 'VIN' (26536475894875647), 'Marca' (Toyota), 'Modelo' (Yaris), 'Quota-Parte' (100%), and 'Declarante' (Preencha com o nome da pessoa). There are three red triangles on the right side of the form, indicating errors. At the bottom, there is a footer with '© 2021-22 - Direcção Geral dos Registos e Notariado' and 'Universidade de Aveiro'.

Figura 70 - Página para validação de um registo

Livro de Registos [AU-001]

Insira um número de registo

Tipo de livro: AU Estado: --- Escolher

Página: 1 de 1
Entradas: 1-6 de 6

N.º Registo	Folha(s)	Tipo de registo	Estado	Última acção	Realizada em	Realizada por	
1		Livro Único Inscrição	○	Catálogoção	Hoje, 13:10:31	Sofia Moniz	A ✓ [ícone]
114		Livro Único Inscrição	○	Edição	Hoje, 16:45:25	Sofia Moniz	A ✓ [ícone]
114/1		Livro Único Averbamento à Inscrição	○	Catálogoção	Hoje, 13:08:21	Sofia Moniz	✓ [ícone]
255		Livro Único Registo	●	Validação	Hoje, 17:51:39	Sofia Moniz	A ✓ [ícone]
255/1		Livro Único Averbamento ao Registo	○	Catálogoção	Hoje, 13:09:51	Sofia Moniz	✓ [ícone]
271		Livro Único Descrição	○	Edição	Hoje, 16:45:24	Sofia Moniz	Z ✓ [ícone]

Simbologia dos ícones e código de cores:
 ○ - Registo catalogado. ● - Registo editado.
 ● - Registo validado. ○ - Registo bloqueado.
 ● - Registo inválido. [ícone] - Notas.

111 ms

Informação 17:51 X
Validação do registo realizada com sucesso.

© 2021-22 - Direcção Geral dos Registos e Notariado Universidade de Aveiro

Figura 71 - Registo 255 validado

Livro de Registos [AU-001]

Insira um número de registo

Tipo de livro: AU Estado: --- Escolher

Página: 1 de 1
Entradas: 1-6 de 6

N.º Registo	Folha(s)	Tipo de registo	Estado	Última acção	Realizada em	Realizada por	
1		Livro Único Inscrição	○	Catálogoção	Hoje, 13:10:31	Sofia Moniz	A ✓ [ícone]
114		Livro Único Inscrição	○	Edição	Hoje, 17:52:11	Sofia Moniz	A ✓ [ícone]
114/1		Livro Único Averbamento à Inscrição	○	Catálogoção	Hoje, 13:08:21	Sofia Moniz	✓ [ícone]
255		Livro Único Registo	●	Edição	Hoje, 17:52:11	Sofia Moniz	A ✓ [ícone]
255/1		Livro Único Averbamento ao Registo	○	Catálogoção	Hoje, 13:09:51	Sofia Moniz	✓ [ícone]
271		Livro Único Descrição	○	Edição	Hoje, 17:52:11	Sofia Moniz	Z ✓ [ícone]

Simbologia dos ícones e código de cores:
 ○ - Registo catalogado. ● - Registo editado.
 ● - Registo validado. ○ - Registo bloqueado.
 ● - Registo inválido. [ícone] - Notas.

89 ms

© 2021-22 - Direcção Geral dos Registos e Notariado Universidade de Aveiro

Figura 72 - Registo 255, após edição

SIRP Registo Predial Registo Automóvel Catalogação Manutenção Administração

sofiamoniz@ua.pt

Registo - Apagar

Tipo de Registo	Livro Único Averbamento à Inscrição		
Data de registo	2022-05-24		
Requerente	Nome	Zé	
	Morada	Zé	
Sujeito(s) activo(s)	Nome	Zé	
	Morada	Zé	
	Estado Civil	Solteiro(a)	
	Quota	100	
Sujeito(s) passivo(s)	Nome	Zé	
	Morada	Zé	
	Estado Civil	Solteiro(a)	
	Quota	100	
Estado do Registo	<input type="radio"/>		
Imagem(s)			
Histórico	Catalogação	24/05/2022 13:08:21	Sofia Moniz
Registos Ascendentes	<input type="checkbox"/> Livro Único Inscrição: 114		

Tem a certeza que pretende apagar este registo?

© 2021-22 - Direcção Geral dos Registos e Notariado Universidade de Aveiro

Figura 73 - Página para eliminação de um registo

ANEXO 6 – RELATÓRIO DE COBERTURA DE TESTES DE INTEGRAÇÃO

No sexto anexo é mostrado, ao detalhe, o relatório gerado através do *Fine Code Coverage*, relativo à cobertura de código através de testes de integração aos controladores da solução.

Generated on:	25/05/2022 - 17:52:47
Parser:	CoberturaParser
Assemblies:	4
Classes:	229
Files:	223
Covered lines:	9128
Uncovered lines:	1146
Coverable lines:	10274
Total lines:	16432
Line coverage:	88.8% (9128 of 10274)
Covered branches:	928
Total branches:	1502
Branch coverage:	61.7% (928 of 1502)

Figura 74 - Relatório gerado através do *Fine Code Coverage*, relativo à cobertura de código (parte 1)

▼ Name	▼ Covered	▼ Uncovered	▼ Coverable	▼ Total	▼ Line coverage	▼ Covered	▼ Total	▼ Branch coverage
+ Microsoft.Owin	0	0	0	0		0	0	
+ Moq	0	0	0	0		0	0	
- WebSite	3428	1097	4525	7387	75.7%	809	1358	59.5%
WebSite.Controllers.AspNetRolesController	68	0	68	135	100%	16	16	100%
WebSite.Controllers.AspNetUsersController	111	36	147	274	75.5%	20	28	71.4%
WebSite.Controllers.AutoController	854	118	972	1286	87.8%	134	204	65.6%
WebSite.Controllers.BuildingsController	206	443	649	881	31.7%	23	162	14.1%
WebSite.Controllers.CatalogingController	204	34	238	299	85.7%	73	110	66.3%
WebSite.Controllers.CertificatesController	0	200	200	331	0%	0	64	0%
WebSite.Controllers.CivilStatesController	69	0	69	143	100%	24	32	75%
WebSite.Controllers.FieldTypesController	73	0	73	147	100%	26	34	76.4%
WebSite.Controllers.FieldsController	96	0	96	170	100%	35	48	72.9%
WebSite.Controllers.FinderController	179	9	188	275	95.2%	60	82	73.1%
WebSite.Controllers.HomeController	30	0	30	68	100%	0	0	
WebSite.Controllers.ImageLogsController	45	0	45	80	100%	19	26	73%
WebSite.Controllers.ImagesController	254	25	279	394	91%	82	114	71.9%
WebSite.Controllers.PersonalController	102	0	102	136	100%	36	38	94.7%
WebSite.Controllers.RegisterLogsController	50	0	50	92	100%	19	26	73%
WebSite.Controllers.RegistersController	430	112	542	941	79.3%	98	154	63.6%
WebSite.Controllers.SearchController	12	47	59	90	20.3%	3	12	25%
WebSite.Controllers.SearchLogsController	77	18	95	165	81%	23	40	57.5%
WebSite.Controllers.SubRegistersController	117	17	134	191	87.3%	17	28	60.7%
WebSite.Controllers.TemplateFieldsController	59	9	68	124	86.7%	19	28	67.8%
WebSite.Controllers.TemplateTypesController	73	0	73	147	100%	28	36	77.7%
WebSite.Controllers.TemplatesController	97	0	97	216	100%	32	44	72.7%
WebSite.Controllers.UsersInRolesController	142	0	142	217	100%	22	32	68.7%

Figura 75 - Relatório gerado através do Fine Code Coverage, relativo à cobertura de código (parte 2)