



Universidade do Minho

Escola de Engenharia

Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Letivo de 2020/2021

Peneda-Vês

João Correia A84414

Marco Pereira A89556

Pedro António A58062

Rúben Cerqueira A89593

10 de maio de 2021

LI4

Data de Receção	
Responsável	
Avaliação	
Observações	

Peneda-Vês

João Correia A84414
Marco Pereira A89556
Pedro António A58062
Rúben Cerqueira A89593

10 de maio de 2021

Resumo

O presente relatório foi elaborado no âmbito do desenvolvimento de um sistema de monitorização e notificação de eventos relativos à fauna do Parque Nacional Peneda-Gerês. Este figura-se como o único Parque Nacional em Portugal, revelando-se a proteção da biosfera nele presente como um tópico de elevada importância.

No curso deste relatório, relativamente à primeira fase de desenvolvimento, serão explorados nove tópicos relativos à especificação do sistema a desenvolver, cobrindo aspetos relativos tanto à justificação da existência e propósito do projeto como aos detalhes do seu planeamento e desenvolvimento.

Serão, depois, no âmbito da segunda fase, explorados quatro tópicos relativos à especificação formal do projeto, onde é descrito o planeamento do desenvolvimento da aplicação, apresentando os requisitos levantados e deles derivando a estrutura da aplicação, o esquema da base de dados e ainda alguns mockups da aplicação.

O primeiro tópico cobre a contextualização do projeto, apresentando o paradigma atual da fauna do Parque Nacional da Peneda-Gerês.

Seguidamente explicita-se a motivação e objetivos do sistema de monitorização a desenvolver, mencionando a necessidade de preservação da vida animal presente no Parque.

Passando à justificação do sistema, reitera-se a ideia da preservação da vida animal, salientando os vários benefícios da implementação do mesmo, tanto em termos ambientais como económicos.

No tópico da utilidade do sistema é detalhado o funcionamento do mesmo. Este estará assente na informação recolhida por várias câmaras, colocadas estrategicamente ao longo do Parque. As entidades responsáveis pela manutenção do Parque serão notificadas da passagem de animais ou seres humanos nas localizações em que as câmaras se encontram instaladas. Dessa informação poderam-se extrair conclusões acerca do estado da vida animal, e também da presença de seres humanos em locais onde a sua presença não é permitida.

No quinto tópico, estabelecimento da identidade do projeto, é revelado o nome do sistema "Peneda-Vês". São apresentados detalhes relativos à existência do projeto, o seu gestor, onde estão estipuladas as condições da sua elaboração, a duração de desenvolvimento, bem como o seu propósito.

De seguida são identificados os recursos necessários ao desenvolvimento e funcionamento do projeto, tanto em termos de infraestrutura tecnológica que suporta o fluxo de informação, como às ferramentas utilizadas pela equipa para planear e desenvolver o sistema.

Será elaborado um esquema representativo do funcionamento do sistema que o projeto envisa implementar, ilustrando os fluxos de informação desde o recolhimento desta até à apresentação ao utilizador.

No tópico da definição de um conjunto de medidas de sucesso são apresentadas métricas e objetivos que a equipa almeja atingir. Estes referem-se tanto a metas de implementação de funcionalidades, como também metas do impacto positivo esperado pela aplicação, como a redução do impacto negativo causado pela entrada de pessoal não autorizado em zonas de acesso restrito.

Posteriormente, é apresentado o plano de desenvolvimento do sistema, sob a forma de um diagrama de Gantt. Neste são descritas as várias etapas necessárias à criação da aplicação, assim como a distribuição da responsabilidade pelos membros da equipa responsável pelo projeto.

Concluída a secção do planeamento do projeto, prossegue-se à segunda fase do projeto, que consiste na especificação do mesmo. Nesta fase são apresentados, em primeiro lugar, os requisitos levantados relativos ao projeto. Estes estarão divididos segundo os vários domínios da aplicação e definem as regras e funcionalidades que o produto final deverá ter.

A partir destes requisitos é desenvolvida, na seguinte secção, a especificação UML do sistema de software. Nesta, é apresentado o modelo de domínio, que representa, de forma simples, o relacionamento entre as várias entidades presentes no sistema. De seguida é apresentado o diagrama de use cases, identificando cada use case e o ator por ele responsável. Estes use cases são depois descritos ao pormenor. Por fim, ainda no contexto desta secção, é apresentado o diagrama de pacotes. Este representa uma componente mais técnica do projeto, onde já são identificados os vários componentes a desenvolver dentro da lógica do ASP.NET.

De seguida, é projetado o sistema de base de dados, sendo identificadas as várias entidades e relacionamentos. Os atributos são caracterizados, culminando com a apresentação do modelo lógico. Por fim é apresentada uma estimativa do espaço em disco da base de dados e a sua taxa de crescimento anual.

Por fim, dentro desta fase, demonstra-se os mockups desenhados para a interface gráfica da aplicação, bem como a sua justificação e fluxo para se conseguir obter o desejado.

Esta fundamentação e especificação servirá de base estrutural à seguinte, e última, fase de implementação do projeto.

Área de Aplicação: Engenharia de Software, Programação Web, Desenho e Arquitectura de Sistemas de Bases de Dados, Sistemas de Monitorização e Notificação.

Palavras-Chave: Planeamento de software, Modelação de sistemas, Unified Modelling Language, Bases de Dados Relacionais, SQL Server, Web, Plataforma .Net Core, C#, ASP.Net, Desenvolvimento Backend, Desenvolvimento Frontend.

Índice

1	Introdução	1
2	Contextualização	2
3	Motivação e objetivos	3
4	Justificação do sistema	4
5	Utilidade do sistema	5
6	Estabelecimento da identidade do projeto	6
7	Maquete do Projeto	7
8	Identificação dos recursos necessários	9
9	Definição de um conjunto de medidas de sucesso	10
10	Plano de desenvolvimento (Diagrama de GANTT)	11
11	Levantamento e Análise de Requisitos	12
11.1	Requisitos Propostos	12
11.1.1	Requisitos de Avistamentos	12
11.1.2	Requisitos de Gestão de Utilizadores	13
11.1.3	Requisitos de Definições de Utilizador	14
11.1.4	Requisitos do Dashboard Principal	14
11.1.5	Requisitos de Administração	15
11.1.6	Requisitos de Gerador de Dados	15
11.2	Análise de Requisitos	16
12	Especificação UML do Sistema de Software a Desenvolver	17
12.1	Modelo de Domínio	17
12.2	Diagrama de Use Cases	18
12.3	Use Cases	19
12.4	Use Cases do Utilizador Base	20
12.4.1	Use Cases do Administrador	22
12.4.2	Use Cases do Utilizador Root	23
12.4.3	Use Cases do Gerador de Dados	23

12.4.4 Diagramas de Atividade	25
12.5 Diagrama de pacotes	27
13 Projeto do sistema de base de dados	30
13.1 Identificação e Caracterização das Entidades a partir de Requisitos	30
13.2 Identificação e Caracterização das Entidades derivadas pela framework Web . .	31
13.3 Identificação e Caracterização dos Relacionamentos	33
13.4 Dicionário de dados	35
13.5 Domínio dos atributos	37
13.6 Modelo lógico	40
13.7 Estimativa do Espaço em Disco da Base de Dados e Taxa de Crescimento Anual	41
13.7.1 Previsão do tamanho inicial	42
13.7.2 Previsão da taxa de crescimento anual	42
14 Definição e caracterização do sistema de interface (mockups)	44
14.1 Dashboard	44
14.2 Página de uma câmara	45
14.3 Página de uma espécie	47
15 Conclusões e Trabalho Futuro	49
Lista de Siglas e Acrónimos	51
Anexos	52

Lista de Figuras

7.1	Maquete do projeto.	7
10.1	Diagrama de GANTT do projeto.	11
12.1	Modelo de Domínio da Aplicação	18
12.2	Diagrama de Use Cases	19
12.3	Diagrama de Atividade para Utilizar Dashboard	25
12.4	Diagrama de Atividade para Registrar Avistamento	26
12.5	Diagrama de Pacotes.	29
13.1	Modelo lógico da base de dados.	40
14.1	Mockup do Dashboard	45
14.2	Mockup da página de uma câmara	46
14.3	Mockup da página Espécies	48
15.1	Utilizar Dashboard Principal	52
15.2	Consultar Espécies	53
15.3	Consultar Câmara	53
15.4	Consultar Câmara	54
15.5	Receber Resumo Diário	54
15.6	Autenticar utilizador	54
15.7	Logout	55
15.8	Logout	55
15.9	Logout	55
15.10	Adicionar Câmara	56
15.11	Adicionar Espécie	56
15.12	Registrar Avistamento	57
15.13	Inicializar o gerador de dados	57
15.14	Notificar Utilizadores de Avistamento	58
15.15	Notificar Administradores de situação de perigo	58
15.16	Gerir Utilizadores	58

Lista de Tabelas

13.1	Dicionário de dados	36
13.2	Dicionário de dados de entidades não utilizadas	36
13.3	Caracterização dos atributos de Camera	37
13.4	Caracterização dos atributos de Species	37
13.5	Caracterização dos atributos de Sighting	38
13.6	Caracterização dos atributos de Utilizador (AspNetUsers	38
13.7	Caracterização dos atributos de AspNetRoles	39
13.8	Caracterização dos atributos de _EFMigrationHistory	39

1 Introdução

Ao longo deste relatório iremos documentar a criação de um sistema de monitorização de eventos, sendo este o projeto da Unidade Curricular de Laboratórios de Informática IV.

O tema escolhido foi a monitorização de um Parque Nacional, uma área protegida para preservar a fauna e flora presentes. Os alertas dados serão alertas de avistamento de espécies, através de câmaras devidamente colocadas ao longo do Parque. As câmaras deverão alertar sempre que for detetado movimento. Estes alertas serão enviados para a nossa aplicação, a qual os irá apresentar numa página web, indicando o local onde ocorreram e a espécie identificada.

O Parque Nacional que decidimos usar para o efeito é o Parque Nacional Peneda-Gerês. Este Parque está maioritariamente situado em Portugal e abriga cerca de 226 espécies de vertebrados, destas, 65 pertencem ao Livro Vermelho de Portugal. Algumas das espécies na região, como a Águia-real (*Aquila Chrysaetos*) e o Lince-Ibérico (*Lynx Pardinus*), encontram-se em estado crítico e há assim uma necessidade de as preservar.

2 Contextualização

O mundo em que vivemos encontra-se num rápido declínio de biodiversidade devido à sobreexploração humana. Como tal torna-se necessária a criação de espaços para a proteção da fauna e flora, estes espaços são muitas vezes Parques nacionais dos respetivos países.

Um exemplo de um destes Parques é o Parque Nacional da Peneda-Gerês, este local é uma área protegida servindo de habitat a muitas espécies de plantas, fungos e animais. Algumas destas encontram-se em estados vulneráveis como o Lobo (*Cannis Lupus*) e o Morcego-de-ferradura-grande (*Rhinolophus Ferrumequinum*).

Embora a área onde estes animais se encontram seja protegida, existem caçadores que gostam de passar o seu tempo livre no desporto da caça, e que devido ao estado vulnerável de alguns destes animais, que acarreta um elevado valor dos seus conteúdos, quebram as regras continuando a caça incessante.

Isto leva à necessidade da existência de uma forma de monitorizar estas zonas de modo a conseguir perceber, qualitativamente, a evolução da população de cada espécie.

Ao longo do nosso trabalho iremos documentar a construção de um sistema de monitorização de eventos que irá permitir ao Parque Nacional da Peneda-Gerês controlar as populações dos vários animais existentes na serra.

O nosso sistema tirará proveito da existência de câmaras espalhadas ao longo das várias zonas do Parque Nacional da Peneda-Gerês. Através de imagens recolhidas em ocasiões de movimentação. As câmaras serão capazes de identificar as espécies presentes na fotografia e irão alertar o sistema da presença daquela espécie próximo do local onde foi fotografada.

3 Motivação e objetivos

Atendendo à enorme biodiversidade da fauna presente no Parque Nacional da Peneda-Gerês foi despertada uma necessidade de vigilância dos seres vivos que lá habitam. Esta monitorização é importante de forma a que permita a preservação da vida animal que aí habita bem como a conservação da biodiversidade atualmente existente do Parque.

Logo, surgiu a ideia de construir uma aplicação capaz de recolher informações que advêm de reconhecimentos fotográficos de animais recolhidos por diferentes câmaras distribuídas pelo Parque.

Esta aplicação, além da possibilidade de reconhecer os animais e humanos nas dadas posições, também poderá analisar o tipo de animal reconhecido e verificar se não existe anomalias na posição atual, e, caso essa situação aconteça, alertar essa anomalia, tomando as medidas indicadas para a tentar contornar, tal como envio de uma equipa especializada de controlo para o local.

4 Justificação do sistema

O presente sistema de rastreio da fauna do Parque Peneda-Gerês tem em vista melhorar o conhecimento existente das espécies encontradas no Parque, assim como a prevenção de acidentes tanto com visitantes, como com animais.

Acredita-se que o investimento atual no Parque não supre as necessidades de segurança para os visitantes, podendo estes perder-se, cruzar-se com espécies perigosas na sua estadia, ou mesmo entrar em áreas interditas.

Esta situação, conjugada com a necessidade de proteção de espécies em vias de extinção, leva a que seja imperativo obter uma melhor visão global do funcionamento do Parque, e acreditamos que a melhor maneira de a obter é através da vigilância, recorrendo a câmaras com capacidade de reconhecimento.

O uso do sistema permitirá garantir que as condições dos animais são óptimas, o que inerentemente tratá consigo investimento exterior, para que se possa estender este rastreio a nível Nacional.

Tal aumento de segurança propiciará também o turismo. Este fluxo de pessoas promoverá o comércio local, bem como o desenvolvimento da região. O investimento exterior para a proteção das espécies, aliado a este fluxo alargado de visitas, compensará assim o investimento inicial para a implementação do sistema.

5 Utilidade do sistema

As câmaras presentes no Parque captarão imagens, onde será identificável a presença de entidades. Estas entidades tanto poderão ser os animais que compõe a fauna do Parque Nacional, como seres humanos. Cada câmara enviará a informação recolhida ao sistema que a irá processar de forma a extrair informação útil à utilização da aplicação.

A aplicação disponibilizará ao utilizador a informação recolhida por cada câmara assim como as entidades detetadas. Sendo o público-alvo do nosso sistema os indivíduos responsáveis pela manutenção e salvaguarda do Parque, o tratamento da informação variará consoante o tipo de entidade que foi detetada.

Caso as entidades detetadas sejam animais, a aplicação indicará também a sua espécie e localização. Esta informação permitirá aos gestores do Parque ter uma melhor ideia sobre a movimentação e localização atual de várias espécies, auxiliando na deteção de situações anormais no comportamento dos animais.

O tratamento da deteção de seres humanos será também útil aos gestores do Parque, permitindo assinalar a presença de pessoal não autorizado em secções reservadas do Parque. Estas situações tanto poderão dever-se a visitantes que se encontram em situações de perigo por se terem perdido, como também se poderão referir a entidades com intenções nocivas, como a caça. Assim, a deteção de humanos concretizar-se-á tanto numa ferramenta capaz de assinalar situações de necessidade de resgate de visitantes, como de invasões ao Parque, reforçando os esforços de conservação deste.

Os utilizadores poderão optar por ser notificados via email e/ou sms de certos eventos que sejam registados pela aplicação.

6 Estabelecimento da identidade do projeto

O projeto deverá ter uma identidade própria que o estabeleça como uma entidade concreta que tenha como propósito criar valor real e não apenas servir como exercício académico de conhecimento. Para tal, a equipa decidiu atribuir ao projeto o nome Peneda-Vês, de forma a realçar a utilidade da aplicação na visualização e monitorização da fauna presente no Parque Nacional da Peneda-Gerês.

Não obstante, é necessário salientar que o projeto se encontra inserido no contexto da unidade curricular Laboratórios de Informática IV, obedecendo às especificações ditadas no enunciado de trabalho. Este enunciado poderá ser, pois, visto como um vínculo contratual entre os membros do projeto e a equipa docente da unidade curricular, onde são declarados quais os objetivos para o desenvolvimento correto do projeto.

O coordenador do grupo de trabalho será o Pedro António, assumindo a função de mediador de comunicação entre o grupo e a equipa docente.

Como referido na justificação do nome do projeto, este terá como finalidade o desenvolvimento de uma aplicação de monitorização da fauna presente no Parque Nacional da Peneda-Gerês, notificando os utilizadores de avistamentos de vários animais, assim como outras informações relevantes à proteção do Parque, e.g. avistamentos de pessoal não autorizado em zonas reservadas do Parque.

O projeto terá uma duração de desenvolvimento de 3 meses, ao fim dos quais se pretende apresentar uma versão final que cumpra todos os requisitos com elevada satisfação. Essa versão final concretizar-se-á na disponibilização de um web site ao qual as entidades competentes à gestão do Parque Nacional poderão aceder para usufruir dos serviços disponibilizados.

7 Maquete do Projeto

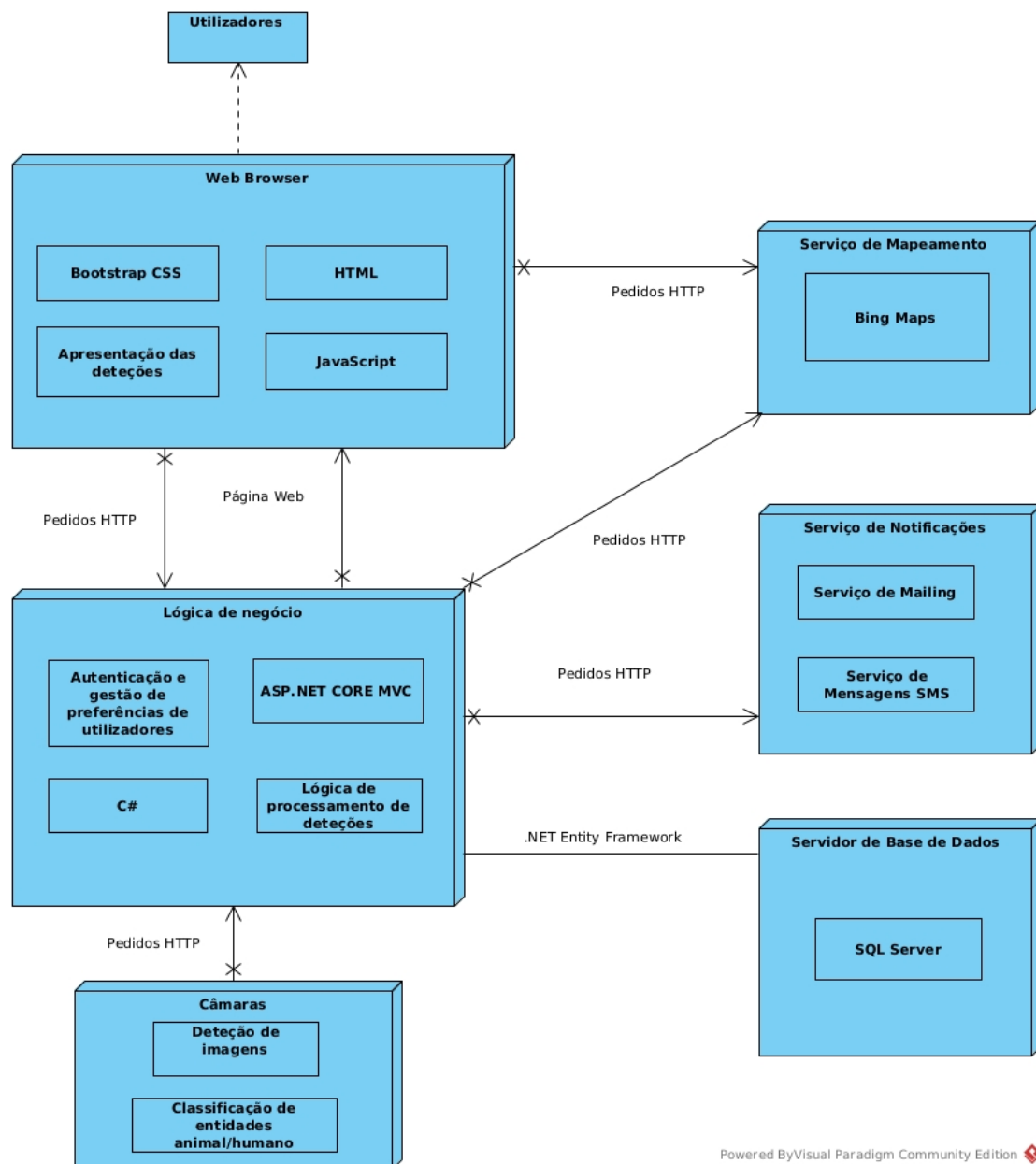


Figura 7.1: Maquete do projeto.

Na base do sistema encontram-se as câmaras espalhadas pelo Parque. Estas serão capazes de detetar imagens que, posteriormente, irão analisar e verificar a presença de animais, cujas espécies identificará, ou seres humanos. As câmaras irão enviar, à lógica de negócio, a informação acerca das entidades detetadas através de pedidos HTTP.

A lógica de negócio, construída com recurso à linguagem de programação C# e utilizando a framework web ASP.NET Core MVC, irá ser responsável por processar a informação enviada pelas câmaras. Esta camada será também responsável por gerir a lógica de utilizadores, permitindo a estes registarem-se e autenticarem-se. Também lhes permite alterar as suas preferências relativas a quais animais pretendem acompanhar e ser notificados. Para tal, a lógica de negócio necessitará de comunicar com serviços de notificações, nomeadamente serviços que permitam enviar notificações via email e SMS.

Um servidor de Base de Dados, SQL Server em particular, permitirá à lógica de negócio armazenar e aceder à informação da aplicação, tanto relativa aos utilizadores, como ao histórico de deteções de entidades. A comunicação entre a lógica de negócio e a base de dados será mediada pela .NET Entity Framework.

Também para processamento da informação georreferenciada enviada pelas câmaras, será utilizada a API disponibilizada pelo Bing Maps.

A componente relativa ao frontend será desenvolvida na stack web comum: HTML, Javascript e CSS, recorrendo, para este último, à framework Bootstrap, utilizada pelo ASP.NET. Esta camada será responsável por apresentar a informação das deteções de animais e humanos ao utilizador, apresentando notificações acerca de eventos importantes.

8 Identificação dos recursos necessários

Poder-se-á dividir os recursos necessários ao desenvolvimento do projeto em duas categorias diferentes.

A primeira categoria corresponde às ferramentas que asseguram o funcionamento do projeto, isto é, componentes tecnológicas que permitem o fluxo de informação dentro da aplicação.

A esta, correspondem, em primeiro lugar, as câmaras colocadas em localizações estratégicas no Parque Nacional. Estas comunicarão com outra componente do projeto, uma aplicação web que disponibilizará *endpoints* aos quais as câmaras poderão enviar a informação recolhida. Esta aplicação será desenvolvida em *ASP.NET*, uma framework web construída sobre a plataforma *.Net Core*. Como motor de gestão de bases de dados será utilizado o *SQL Server*. Recorrer-se-á a serviços externos de notificação do utilizador, para envio de emails e mensagens sms. Será ainda utilizada a API disponibilizada pelo Bing Maps na apresentação da informação georeferenciada ao utilizador.

A segunda componente corresponde a ferramentas associadas ao planeamento e desenvolvimento do projeto, sem terem interação direta com o utilizador.

A linguagem de modelação UML será uma das componentes que se insere nesta categoria, sendo utilizada no desenvolvimento de vários diagramas ilustrativos do processo de desenvolvimento e do comportamento do programa. Estes diagramas serão desenvolvidos com o auxílio do software Visual Paradigm.

Também a plataforma Github verá o seu papel inserido na segunda categoria, sendo utilizada com o propósito de servir como repositório de código, facilitando assim a colaboração da equipa.

9 Definição de um conjunto de medidas de sucesso

No intuito de existirem métricas de avaliação para que o projecto seja considerado um sucesso, foram estipulados os seguintes objectivos:

- O desenvolvimento de toda a estrutura de rastreio deverá ser mantido dentro dos parâmetros de financiamento estabelecidos.
- Permitir a implementação e o suporte de pelo menos 30 câmaras espalhadas pelo Parque. Acredita-se que esta quantidade de câmaras permitirá o rastreio de uma quantidade já significativa da fauna, levando ao interesse em investimento externo e ao começo de um sistema de rastreio a nível Nacional.
- Desenvolver um sistema capaz de processar, no mínimo, 85% dos alertas recebidos.
- A criação de uma base de dados capaz de armazenar as informações recolhidas pelas câmaras, servindo depois como suporte à aplicação desenvolvida.
- Permitir ao utilizador da aplicação seguir os movimentos de espécies específicas, assim como obter alertas de movimentações nas câmaras que pretender seguir.
- Reduzir drasticamente acidentes, encontros com animais perigosos, assim como entrada de pessoas em áreas restritas.
- Produzir atualizações com um espaçamento temporal conducente a uma ideia bem fundamentada do funcionamento do Parque.
- Durante a fase de planeamento, estipulou-se que o processo de desenvolvimento teria a duração de 3 meses, e deverá ser entregue, numa configuração que obedeça a todas as metas citadas acima, até ao dia 7 de Junho de 2021.

10 Plano de desenvolvimento (Diagrama de GANTT)

No início de cada projeto, é importante definir as bases do mesmo antes de começar a construção, estabelecendo objetivos, funções e esqueleto deste.

Mas tão importante quanto aos pontos referidos é também planejar o tempo dedicado à concretização das diferentes etapas de modo a que se consiga apresentar um produto final com qualidade e que cumpra os prazos estabelecidos.

Para ajudar neste planeamento recorremos ao diagrama de Gantt que nos permite demonstrar graficamente o esquema. O diagrama construído para o presente projeto é o seguinte:

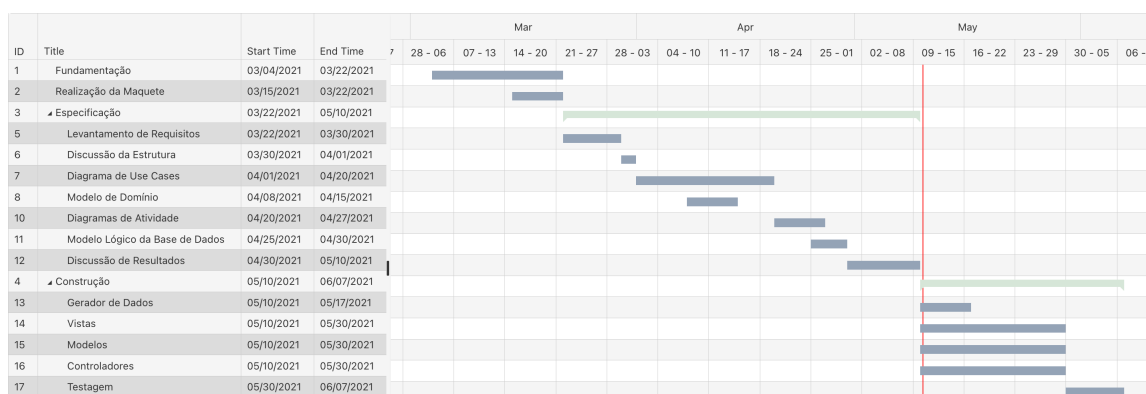


Figura 10.1: Diagrama de GANTT do projeto.

11 Levantamento e Análise de Requisitos

O levantamento de requisitos é uma das partes mais importantes no planeamento de desenvolvimento de software. É nesta fase que se estabelecem as metas a atingir para o sucesso da aplicação.

Uma vez que não houve um cliente com o qual acordar os requisitos propostos, houve uma avaliação e reiteração minuciosa realizadas pela equipa de desenvolvimento de forma a que o desenvolvimento da aplicação tenha base coesa, procurando evitar ambiguidades nos objetivos traçados e eficazmente estabelecer as regras de produção para a concretização de um produto completo e estável.

11.1 Requisitos Propostos

Enumeram-se, de seguida, os vários requisitos levantados pela equipa, separados por categoria.

11.1.1 Requisitos de Avistamentos

- RAv 1. Um espécie é identificada pelo seu id, nome comum, nome científico, uma descrição, se uma espécie tem características predatórias, assim como uma imagem;
- RAv 2. Uma câmara é caracterizada pelo seu id, nome, a sua localização geográfica (latitude e longitude), se se encontra numa zona restrita do parque e ainda se é acessível apenas por administradores ou também pelos utilizadores base;
- RAv 3. Cada utilizador relaciona-se com as câmaras que pretende visualizar;
- RAv 4. Cada utilizador relaciona-se com as espécies que pretende seguir;
- RAv 5. Um avistamento é caracterizado pela sua data e hora de deteção, assim como o número de indivíduos detectados;
- RAv 6. Um avistamento relaciona-se com a câmara que o realizou e a espécie detetada;

- RAv 7. Apenas é permitida uma câmara e uma espécie por avistamento;
- RAv 8. A aplicação web disponibiliza um endpoint no qual recebe registos de avistamentos, indicando o id da câmara que o detetou, o id da espécie detetada e quantos indivíduos da espécie foram detetados;
- RAv 9. Quando a aplicação regista uma nova deteção, insere-a na base de dados e verifica todos os utilizadores que indicaram que pretendem receber notificação dessa espécie e câmara, enviando uma notificação no browser do utilizador;
- RAv 10. Caso a espécie detetada seja um ou mais seres humanos numa câmara assinalada como estando em área reservada, é enviada uma notificação a cada um dos administradores, sugerindo o envio de uma equipa de atuação. Esta notificação é enviada para o meio de comunicação definido pelo administrador (email e/ou telemóvel);
- RAv 11. Caso a espécie detetada seja um ou mais seres humanos e, nos 3 minutos prévios tenham sido detetados indivíduos de uma espécie com comportamentos predatórios, na mesma câmara, é enviada uma notificação a cada administrador, indicando a zona à qual se deve enviar uma equipa de resgate. Esta notificação é enviada para o browser e para o meio de comunicação definido pelo administrador;
- RAv 12. Caso a espécie detetada seja uma espécie predatória e, nos 3 minutos prévios tenham sido detetados seres humanos, é enviada uma notificação a cada administrador, indicando a zona à qual se deve enviar uma equipa de resgate. Esta notificação é enviada para o meio de comunicação definido pelo administrador;
- RAv 13. O utilizador recebe diariamente, um resumo dos avistamentos que ocorreram nesse intervalo, no meio de comunicação definido pelo próprio (telemóvel e/ou email);
- RAv 14. O envio dos resumos e das notificações de emergência dos administradores para telemóvel/email é feito através da utilização de APIs externas.

11.1.2 Requisitos de Gestão de Utilizadores

- RGU 1. Um utilizador é identificado por um id, email, nome, password, número de telemóvel opcional, se pretende receber o resumo diário e os meios onde o pretende receber (email e/ou telemóvel);
- RGU 2. Um utilizador relaciona-se também com os roles/permisões a ele associados, nomeadamente se é administrador, utilizador base ou ainda o utilizador root;
- RGU 3. A página inicial da aplicação apresenta ao utilizador a opção de login na aplicação, onde este deve inserir email e password;
- RGU 4. A página inicial tem a opção de redirecionar o utilizador para uma página de registo;

- RGU 5. Na página de registo o utilizador deverá preencher o seu email, nome, password, confirmar a password, opcionalmente o número de telemóvel, se pretende receber o resumo diário e, caso afirmativo, em que meios (email ou telemóvel) o pretende receber;
- RGU 6. As contas com permissões de administrador incluem todas as funcionalidades das contas normais, mais funcionalidades específicas de administração do parque;
- RGU 7. A conta root tem todas as permissões de administrador e ainda a funcionalidade de gestão de utilizadores, nomeadamente promoção e demissão do papel de administrador.
- RGU 8. A conta root tem acesso a um painel com informação acerca dos vários utilizadores registados na aplicação;

11.1.3 Requisitos de Definições de Utilizador

- RDU 1. Existe uma página específica onde o utilizador consegue alterar as definições relativas à sua conta;
- RDU 2. O utilizador define/altera o número de telemóvel onde pretende receber o resumo diário;
- RDU 3. O utilizador define de quais câmaras pretende receber notificações de avistamentos;
- RDU 4. O utilizador define de quais espécies pretende receber notificações de avistamentos.

11.1.4 Requisitos do Dashboard Principal

- RDP 1. A página principal da aplicação apresenta as várias câmaras disponibilizadas no parque, indicadas com referência geográfica sobre um mapa do Parque Nacional Peneda-Gerês, obtido a partir do Bing Maps;
- RDP 2. Passando o cursor sobre a câmara, é apresentada metainformação sobre esta: nome, localização e o número de avistamentos registados nos últimos 7 dias;
- RDP 3. A página apresenta, numa secção separada da secção do mapa, os avistamentos mais recentes ordenados cronologicamente, indicando a espécie, câmara, número de indivíduos e momento de deteção;
- RDP 4. Clicando sobre cada câmara, o utilizador é redirecionado para a sua página, onde é apresentado o histórico de avistamentos da câmara delimitado por um intervalo temporal ajustável, assim como informação relativa à câmara: nome, localização e se se encontra numa zona de acesso restrito;
- RDP 5. A página da câmara contém ainda um gráfico do número de avistamentos por unidade temporal calculada a partir do intervalo temporal usado para filtrar os avistamentos;

- RDP 6. A aplicação permite consultar as espécies existentes no parque, abrindo uma nova página onde estas são listadas pelo seu nome e o número de avistamentos;
- RDP 7. Clicar numa das espécies revela uma nova página com mais informação acerca dela: nome comum, nome científico, descrição, se a espécie é predatória e a lista de avistamentos delimitados por um intervalo temporal ajustável, cada um contendo a data e a câmara;
- RDP 8. Também na página da espécie é apresentado um gráfico representativo do número de avistamentos por unidade temporal.

11.1.5 Requisitos de Administração

- RDA 1. Na página de uma câmara é possível alterar informações relativas a essa câmara;
- RDA 2. Quando uma câmara de acesso geral é alterada para acesso restrito pelos administradores, esta deve ser removida das preferências dos utilizadores;
- RDA 3. É possível registar uma nova câmara, indicando a sua localização geográfica (latitude e longitude) e um nome. Define-se também se a câmara está associada a uma zona reservada do parque, e se os utilizadores base da aplicação devem ter acesso aos avistamentos por ela registados;
- RDA 4. É possível adicionar uma espécie, fornecendo nome comum, nome científico, uma descrição, se a espécie tem características predatórias, assim como, uma imagem representativa;
- RDA 5. Permite editar uma espécie existente.

11.1.6 Requisitos de Gerador de Dados

- RGD 1. O gerador de dados será uma aplicação persistente, que ao longo do seu funcionamento irá enviar registos de avistamentos para a aplicação central;
- RGD 2. A aplicação web principal disponibiliza um endpoint que permite ao gerador de dados consultar a lista de câmaras existentes, contendo o ID da câmara assim como a sua localização;
- RGD 3. A aplicação web disponibiliza um endpoint que permite ao gerador de dados consultar a lista de espécies existentes, contendo os seus IDs;
- RGD 4. Para registar novos avistamentos, o gerador de dados deverá enviar um pedido POST para o endpoint pré-definido pela aplicação principal, indicando o id da espécie detetada, o id da câmara que o detetou e o número de indivíduos detetados.

11.2 Análise de Requisitos

Devido ao tema e objetivo do sistema a desenvolver, foi necessário estabelecer várias categorias que remetem para as várias áreas específicas do software a planejar.

Enumeram-se, assim, as categorias levantadas: **Avistamentos**, **Gestão de Utilizadores**, **Definições de Utilizador**, **Dashboard geral**, **Administração** e **Gerador de Dados**.

A primeira categoria remete para a questão dos **avistamentos**, ou seja, os dados com que a aplicação vai trabalhar recolhidos pelas câmaras distribuídas pelo Parque. Estes requisitos delimitam os detalhes da informação recolhida, bem como o fluxo da informação e o comportamento das câmaras.

Seguidamente são tratados os requisitos de **gestão de utilizadores**. São estabelecidos três tipos de utilizadores: **Utilizador Base**, **Administrador** e **Root**. Os requisitos desta categoria enumeram as permissões de cada tipo de conta, bem como as suas informações e características.

Nas **definições de utilizador** definem-se as customizações que cada Utilizador pode realizar na aplicação. Estas customizações aplicam-se às notificações a receber bem como acesso a outro tipo de informações adicionais.

Fixa-se, também, requisitos do **dashboard principal** que caracteriza o software a implementar. As imposições desta categoria maioritariamente detalham o fluxo pretendido da página principal da aplicação, bem como os diferentes filtros que lhe podem ser aplicados.

De seguida são impostos os requisitos referentes à **administração** do sistema, ou seja, apenas associados às contas de administração. Nestes requisitos definem-se as permissões adicionais em que estas contas diferem nas contas base, permitindo um manuseamento superficial do sistema, tal como a gestão de câmaras.

Por fim, são designados as premissas associadas à comunicação com a fonte de dados do sistema. Tais requisitos delineiam o tipo de comunicação, bem como a mesma é realizada.

Com todos os requisitos elaborados, foi possível uma base de orientação para as etapas seguintes do planeamento do projeto e posterior implementação. Esta elaboração permite uma maior organização do espaço de trabalho, sendo possível a eliminação do número máximo possível de ambiguidades que poderiam originar e assim delinear uma aplicação bem elaborada e estável.

12 Especificação UML do Sistema de Software a Desenvolver

12.1 Modelo de Domínio

A partir dos requisitos previamente definidos, é agora possível desenvolver-se um **Modelo de Domínio** para a aplicação a desenvolver.

Um **Modelo de Domínio** é uma representação conceptual que, recorrendo à notação UML, tem em vista demonstrar todas as interações que decorrem no sistema que pretendemos implementar. Este, caracteriza-se pela simplicidade de notação, que o torna ideal para uma apresentação ao cliente, uma vez que mesmo sem conhecimento técnico do funcionamento de uma aplicação, pode rapidamente perceber os relacionamentos entre todas as entidades presentes no sistema.

A figura 12.1 apresenta o **Modelo de Domínio** desenvolvido para a aplicação Peneda-Vês.

Através de uma análise ao **Modelo de Domínio**, pode-se rapidamente identificar todos os componentes envolvidos na construção da aplicação, e que garantem que esta cumpre com todas as necessidades levantadas:

A Aplicação tem uma Dashboard principal, onde se apresenta o Mapa do Parque, contendo todas as câmaras fisicamente instaladas. Estas, estão responsáveis pelo registo de Avistamentos de Espécies, os quais são simulados pelo Gerador de dados.

Existe um Utilizador root, responsável por gerir os Administradores, podendo promover Utilizadores Base a Administradores, ou remover a autorização a estes. Os Administradores podem criar e editar Espécies disponíveis no sistema, assim como receber notificações de emergência, despoletadas por Avistamentos.

O Utilizador Base recebe um Resumo diário contendo os Avistamentos que seleccionou estar interessado em receber através das Preferências. Recorrendo ao Serviço de notificação, o Utilizador Base pode ainda definir como Meio de contacto o seu email, ou SMS.

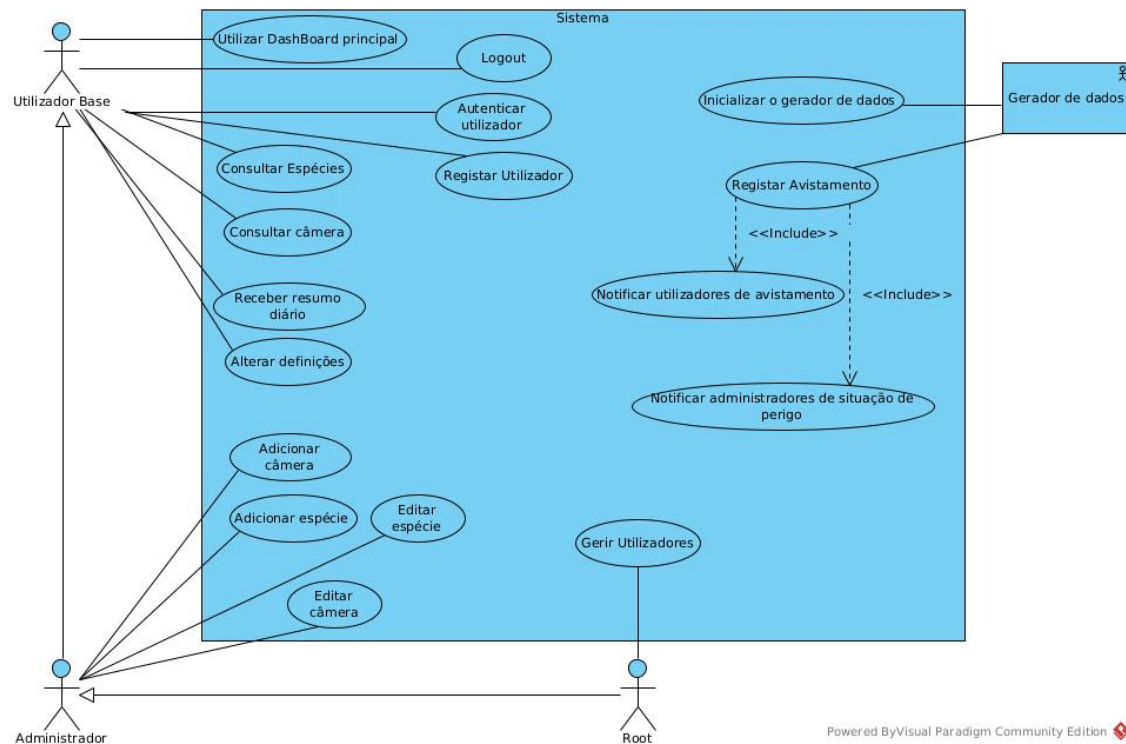


Figura 12.2: Diagrama de Use Cases

tem acesso, e tendo também permissão para adicionar câmaras e espécies ao sistema.

- **Root:** Utilizador que possui acesso a nível "root" à aplicação, podendo promover Utilizadores Base a Administradores, estando responsável pela sua gestão.
- **Gerador de Dados:** Este ator está responsável pela geração de avistamentos, assim como o envio de notificações a todos os utilizadores.

Será agora feita uma análise pormenorizada de cada um dos **Use Cases** constantes do **Diagrama de Use Cases**. Será detalhado todo o fluxo normal, que leva ao funcionamento esperado da aplicação; o fluxo alternativo, que se prende com situações alternativas, mas mesmo assim conducentes a um funcionamento normal; e também possíveis exceções, que correspondem a situações anómalas e que necessitam de tratamento próprio.

12.3 Use Cases

Descreveremos seguidamente cada um dos Use Cases de forma textual. Para uma consulta mais rápida, estes podem também ser encontrados em anexo, em formato de tabela.

12.4 Use Cases do Utilizador Base

Registar Utilizador

Para que possa utilizar a aplicação, um Utilizador deverá estar registado. Para tal, na área de registo, insere as suas credenciais: email, nome, password, confirmação da password, e opcionalmente o seu número de telemóvel. Seguidamente, indica se pretende receber o resumo diário de avistamentos, seleccionando se pretende receber o mesmo por email, ou por SMS.

O Sistema valida as credenciais, verificando que o email ainda não está registado, e que a password segue os requisitos mínimos, criando um novo registo. Redireciona o Utilizador já autenticado para o dashboard principal.

É ainda gerada uma exceção na eventualidade do Utilizador não inserir credenciais válidas, sendo este informado pelo Sistema.

Autenticar Utilizador

O Utilizador pode efetuar a sua autenticação na aplicação, com a pré-condição de que não esteja já autenticado. Começa por inserir as suas credenciais (endereço de email, e password), que o Sistema seguidamente valida.

Se os dados constarem na Base de Dados, o Utilizador é redirecionado para a página principal, já autenticado. Caso contrário, é gerada uma situação de exceção, em que o Utilizador é informado que as credenciais inseridas não são válidas.

Logout

Estando autenticado, o Utilizador pode escolher terminar a sua sessão na aplicação, seleccionando a opção logout no dashboard principal. A sessão atual é eliminada do Sistema, e o Utilizador é assim redirecionado de novo para o ecrã de login.

Utilizar Dashboard Principal

A utilização do dashboard principal está disponível a Utilizadores autenticados. O Sistema confirma as câmaras que o Utilizador selecionou, e para cada uma delas disponibiliza os avistamentos de espécies seguidas.

É chamada agora a API do Bing Maps, por forma a se obter o mapa do Parque Nacional da Peneda Gerês, no qual o Sistema projeta cada uma das câmaras, de acordo com as coordenadas associadas. Adiciona-se também informação como o nome da câmara, número de avistamentos

detetados num intervalo pré-definido, bem como uma hiperligação clicável na imagem da câmara, que dá acesso à sua lista de avistamentos completa.

Numa zona da vista separada do mapa, o Sistema projeta adicionalmente informação dos avistamentos, de forma cronologicamente descendente, antes de devolver a vista ao Utilizador.

Receber Resumo Diário

O Utilizador pode escolher receber um resumo diário dos avistamentos capturados no dia. Para o efeito, o sistema lança um CRON job diariamente, seleccionando todos os utilizadores que assinalaram a sua preferência por receber o resumo diário.

Para cada utilizador, é recolhida a informação para as câmaras que seleccionaram, assim como as espécies indicadas. É feito um filtro sobre os avistamentos para obter apenas os recolhidos no próprio dia, criando-se um texto de resumo.

Finalmente, o Sistema verifica qual o meio de notificação escolhido por cada Utilizador, antes de contactar a API de notificação externa para que esta entregue o resumo ao Utilizador.

Consultar Câmara

Estando autenticado, um Utilizador pode seleccionar uma câmara no dashboard para consultar.

O processo é o seguinte: o Utilizador selecciona qual a câmara que pretende consultar, ponto em que o Sistema acede à Base de Dados para obter informação sobre a câmara e os avistamentos por esta capturados, podendo o Utilizador definir um intervalo temporal no qual deseja obter avistamentos. Estes são filtrados por forma a apenas apresentar as espécies definidas pelo Utilizador, ordenando-os por ordem cronológica.

A informação é então devolvida, com um número máximo de entradas, definido após testes de performance da aplicação, gerando-se um gráfico do número de avistamentos por unidade temporal.

Consultar Espécies

A fim de consultar as espécies na aplicação, o Utilizador pode solicitar a listagem das espécies ao Sistema. Este, acede à Base de Dados, e seleccionando todas as espécies existentes no Parque, apresenta-as ao Utilizador.

Seguidamente, o Utilizador pode clicar numa espécie, para que o Sistema apresente os seguintes detalhes: nome comum, nome científico, imagem, descrição, e se tem ou não características predatórias. A fim de obter esta informação, o Sistema selecciona os avistamentos da espécie nas câmaras para as quais o Utilizador pretende obter informações, podendo este seleccionar um intervalo temporal para os avistamentos.

Por fim, o Sistema apresenta ao Utilizador os avistamentos da espécie, elaborando também um gráfico do número de avistamentos por unidade temporal.

Alterar Definições

O Utilizador pode alterar as definições associadas à sua conta, clicando na opção disponível no dashboard principal. O Sistema apresenta então uma página dividida em quatro secções: informações da conta, câmaras, e espécies.

Na secção de informações da conta, o Utilizador pode alterar o seu username, se pretende receber o resumo diário, assim como o meio pelo qual o pretende receber (email ou telemóvel). Na área das câmaras, o Utilizador decide quais deseja ver no dashboard principal, assim como quais constam no seu resumo diário.

Finalmente, na secção das espécies, pode seleccionar-se quais as espécies cujos avistamentos interessam ao Utilizador.

12.4.1 Use Cases do Administrador

Editar Câmara

Encontrando-se na página de visualização da câmara, o Administrador pode escolher a opção de a editar. O Sistema apresenta-lhe um formulário contendo os vários campos de informação relativos à câmara, no qual o Administrador altera as definições que desejar, e submete o formulário.

Após recebido o formulário, o Sistema atualiza o registo da câmara na Base de Dados.

Como fluxo alternativo, o Administrador pode tornar o uso da câmara como restrito. Assim que recebe o formulário, o Sistema remove também a câmara das preferências de todos os Utilizadores Base.

Adicionar Câmara

O Administrador pode registar uma nova câmara no sistema seleccionando a opção de adicionar uma nova câmara.

O Sistema apresenta um formulário com os vários campos a preencher: nome, latitude, longitude, restrições de acesso e de localização, que após preenchido pelo Administrador, é validado. Gera-se um fluxo de exceção caso o formulário contenha campos inválidos, e o Administrador é avisado que deve preencher corretamente o formulário.

A câmara recebe um novo id do Sistema, e é inserida na Base de Dados.

Adicionar Espécie

Estando autenticado, um Administrador tem a opção de registar uma nova espécie no sistema. Para o efeito, seleciona a operação de adicionar uma nova espécie, que despoleta a apresentação de um formulário de adição de espécie por parte do Sistema.

O Administrador preenche os vários campos, e submete o formulário. Este é validado pelo Sistema, gerando-se uma exceção caso algum dos campos seja preenchido de forma inválida, sendo pedido ao Administrador que os preencha corretamente, e reenvie o formulário.

Tendo validado o formulário, o Sistema atribui um novo id à espécie criada, e insere-a na Base de Dados.

12.4.2 Use Cases do Utilizador Root

Gerir Administradores

O Utilizador Root pode alterar os acessos de Administradores na aplicação. Para tal, seleciona a página de gestão de Utilizadores. O Sistema apresenta todos os Utilizadores na aplicação, nos quais o Utilizador Root pode clicar para abrir uma janela com os seus detalhes.

Nesta vista, o Utilizador Root pode escolher promover um Utilizador a Administrador, que é processado pelo Sistema e adicionado à Base de Dados. Alternativamente, o Utilizador Root pode também escolher tirar o acesso de Administrador, restringindo as permissões desse Utilizador às de um Utilizador Base.

12.4.3 Use Cases do Gerador de Dados

Inicializar o Gerador de Dados

Para que o Gerador de Dados possa obter informação sobre as espécies e câmaras e gerar avistamentos, este deverá estar inicializado.

O processo começa com o envio de um pedido GET ao endpoint da aplicação responsável por devolver informação sobre as espécies. O Sistema consulta a tabela das espécies, gerando uma lista de pares (Id da espécie, nome comum), convertendo-a para formato JSON, e devolvendo-a como resposta ao pedido do Gerador.

Recebendo esta resposta, o Gerador guarda a informação, e envia um novo pedido GET, desta

vez ao endpoint da aplicação responsável por devolver informação sobre as câmaras existentes no Parque.

O Sistema consulta a tabela na Base de Dados que contém informação relativa às câmaras, gerando uma lista de pares (Id da câmara, nome), convertida também para o formato JSON, e sendo devolvida em resposta ao pedido do Gerador.

Finalmente, o Gerador aceita a resposta no formato JSON com a informação sobre as câmaras existentes no Parque, e guarda-a para que possa conhecer as câmaras para as quais pode gerar avistamentos.

Registar Avistamento

O Gerador de Dados escolhe aleatoriamente um id de uma câmara e de uma espécie, assim como um número de indivíduos avistados. Envia seguidamente um pedido POST ao endpoint da aplicação responsável por registar avistamentos, contendo no corpo do pedido o id da câmara, o id da espécie, assim como o número de indivíduos avistados.

O Sistema recebe o pedido, validando a existência do id da câmara e o id da espécie, e criando um novo registo na tabela dos avistamentos, sendo-lhe atribuído um id e registo temporal, além da informação recebida no pedido. Gera-se um fluxo de exceção caso o id da câmara ou da espécie não existam, sendo enviado um código 404 ao Gerador.

Após criado o registo, o Sistema verifica que o avistamento não se enquadra numa das situações de perigo, e notifica os Utilizadores deste avistamento (ação descrita abaixo, no Use Case **Notificar Utilizadores de Avistamento**). Caso a situação retrate um humano numa área restrita, ou um animal com características predatórias numa zona com humanos, o Sistema notifica os Administradores (ação descrita abaixo, no Use Case **Notificar Administradores de Situação de Perigo**).

Notificar Utilizadores de Avistamento

Depois de receber uma notificação de avistamento por parte do Gerador de Dados, o Sistema seleciona da Base de Dados, os utilizadores que escolheram receber avistamentos desta espécie, e desta câmara. Seguidamente, envia uma notificação push contendo a informação do avistamento.

Notificar Administradores de Situação de Perigo

Quando o Sistema recebe uma notificação de avistamento que corresponde a uma das situações de perigo, envia uma notificação push para os browsers de todos os administradores, indicando a situação de perigo e aconselhando o envio de uma equipa especializada.

Por cada administrador, o sistema verifica o meio de notificação pretendido: email, telefone, ou ambos, recolhendo o seu valor. Finalmente, o Sistema envia pedido aos serviços de notificação externos, com o corpo da mensagem a enviar, assim como o número de telefone ou o endereço de email.

12.4.4 Diagramas de Atividade

Os Diagramas de Atividade são usados para ilustrar o fluxo de controlo num sistema, referindo os passos envolvidos no seu processo. Estes podem modelar atividade sequenciais como concorrentes, e têm uma forte componente visual, sendo bastante úteis para rapidamente se ter uma boa noção do funcionamento do Use Case e de todas as alternativas e estados envolvidos.

Seguidamente serão demonstrados os Diagramas de Atividade desenvolvidos para funcionalidade de **Utilizar Dashboard**, assim como a de **Registar Avistamento**.

Utilizar Dashboard

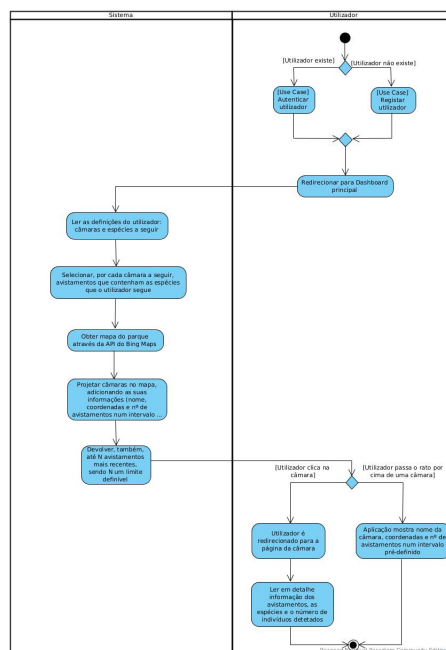


Figura 12.3: Diagrama de Atividade para Utilizar Dashboard

Inicialmente averigua-se se o Utilizador existe, utilizando-se a funcionalidade de registo caso este não exista. Existindo, o Utilizador é autenticado, e redirecionado para a Dashboard principal.

Seguidamente, as definições são lidas, e os avistamentos relevantes ao Utilizador são projetados no mapa, obtido através da API do Bing Maps. Adicionalmente, são devolvidos até N

avistamentos mais recentes, N sendo o limite superior de avistamentos a disponibilizar, obtido através de stress tests à aplicação.

O Utilizador pode agora passar com o rato por cima de uma das câmaras, para o qual a aplicação mostra o nome da câmara, coordenadas, assim como o número de avistamentos recentes. Alternativamente, o Utilizador pode clicar numa das câmaras, sendo redirecionado para a página da câmara.

Na página da câmara, o Utilizador tem acesso à informação detalhada dos avistamentos, incluindo as espécies e o número de entidades avistadas, concluindo assim o fluxo de interação com a Dashboard principal.

Registrar Avistamento

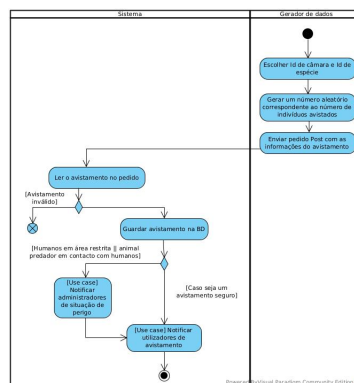


Figura 12.4: Diagrama de Atividade para Registrar Avistamento

Neste diagrama de atividade, detalha-se o fluxo que leva ao registo de um avistamento.

O Gerador de Dados começa por escolher o Id de uma câmara e de um espécie, gerando um número aleatório correspondente ao número de indivíduos avistados. Seguidamente envia um pedido Post com as informações do avistamento ao Sistema.

O Sistema lê o avistamento, e descarta-o caso este seja inválido. De outro modo, guarda-o na Base de Dados. Depois de avaliar o tipo de avistamento, opta por enviar uma notificação aos Administradores se se tratar de uma situação de perigo, e, caso seja apenas uma situação normal, envia a notificação aos Utilizadores Base.

12.5 Diagrama de pacotes

De forma a ilustrar a estruturação das várias componentes do projeto, foi primeiramente idealizado um diagrama de componentes. Porém, dois fatores diferentes levaram-nos a concluir que este não seria o melhor diagrama para representar a aplicação:

- **Injecção de dependências** - O ASP.NET, funciona utilizando a lógica de injeção de dependências e inversão de controlo. Nesta filosofia, um módulo do programa deixa de poder instanciar componentes da lógica de outro módulo, sendo obrigado a recebê-los pelo seu controlador (ex: um controlador que pretenda aceder à base de dados deve receber o módulo de acesso à base de dados pelo seu construtor). Isto leva a que não exista a grande isolação de lógica e falta de partilha de código, características que são normalmente associadas com o padrão de desenho de software por *facades*, padrão este que é melhor representado pelo diagrama de componentes, ao invés do padrão de injeção de dependências, onde existe maior partilha.
- **Interfaces de entrada** - Em seguimento do primeiro ponto, os diagramas de componentes assumem também que cada módulo implementa uma única interface, sendo esse o seu único ponto de acesso. Isto não acontece na injeção de dependências, em que várias componentes do mesmo módulo podem ser expostas, ao invés de apenas o que seria interpretado, normalmente, pela sua *facade*.

Dada a anterior justificação, julgou-se mais elucidativa a apresentação de um **diagrama de pacotes** como forma de representar a organização lógica do projeto.

Este é apresentado na figura 12.5

Este diagrama apresenta 12 diferentes pacotes que serão agora explicados:

- **PenedaVes** - Este é o pacote principal da aplicação. Será nele que todo o código do projeto se encontra inserido, assim como os seus pacotes associados.
- **Controllers** - Este pacote contém os vários controllers da aplicação. Estes são componentes responsáveis por atender os pedidos HTTP provenientes dos vários clientes da aplicação e executar a lógica associada a esses pedidos, interagindo com os outros pacotes, na maioria dos casos.
- **Models** - Este pacote contém a representação, em código, das várias entidades da base de dados.
- **ViewModels** - Este pacote contém modelos temporários, logo não existentes na base de dados, que são normalmente utilizados como uma forma de passar e/ou recolher informação das vistas da aplicação.
- **Configuration** - Contém várias classes que guardam informação útil ao funcionamento

dos restantes pacotes.

- **View** - Contém as várias vistas da aplicação, isto é, as páginas HTML + CSS + Javascript que são apresentadas aos utilizadores quando estes interagem com a aplicação através do browser.
- **Data** - Este contém dois pacotes responsáveis por processar ficheiros e abstrair o acesso à base de dados. Contém também a representação do contexto da base de dados.
 - **FileManager** - Contém a lógica associada ao processamento de ficheiros que são *uploaded* por utilizadores da aplicação. Por exemplo, contém scripts de manipulação das imagens das espécies, quando estas são inseridas pelo utilizador, na criação de uma.
 - **Repository** - Contém e abstrai dados a lógica de acesso à base de dados. É utilizado pelos controladores quando estes pretendem consultar/alterar os dados existentes, ou ainda criar/remover entidades.
- **Services** - Contém dois pacotes responsáveis por abstrair a utilização de API's externas para enviar notificações e o resumo diário.
 - **EmailService** - Abstrai a lógica do envio de mensagens email.
 - **PhoneService** - Abstrai a lógica do envio de mensagens SMS.

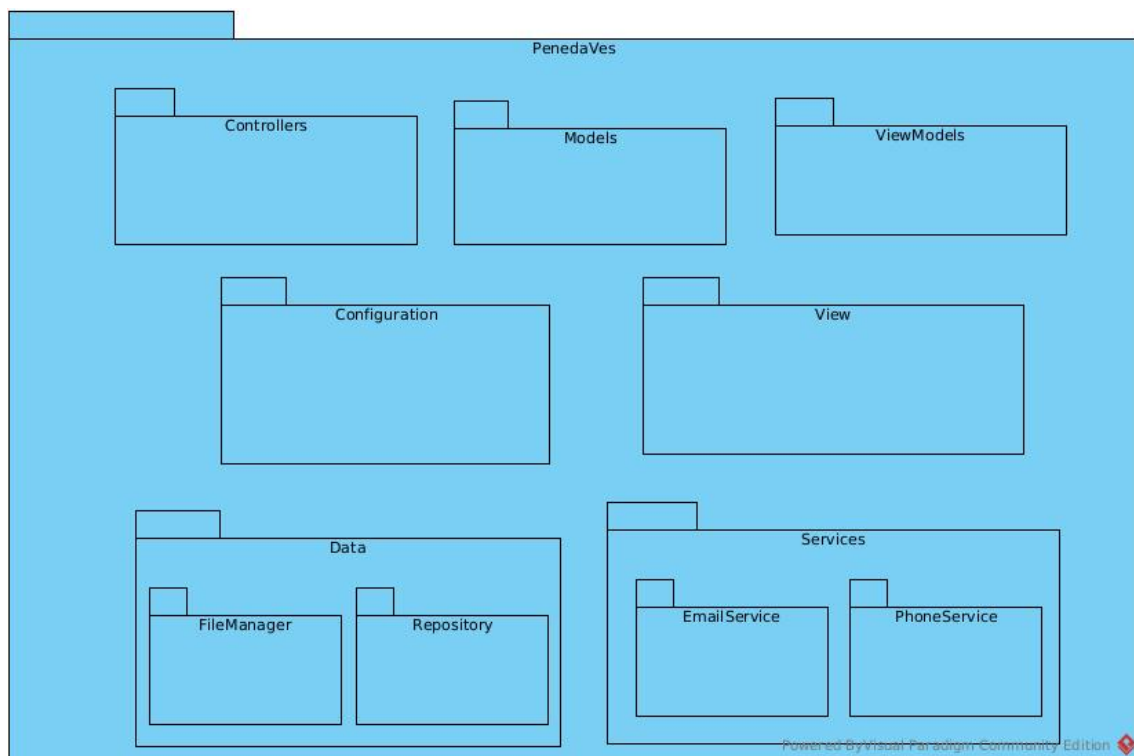


Figura 12.5: Diagrama de Pacotes.

13 Projeto do sistema de base de dados

O desenho e implementação do sistema de base de dados relativo ao projeto Peneda-Vês é um processo caracterizado pela sua submissão a dois vetores: por um lado, este deve satisfazer os requisitos funcionais levantados durante o planeamento do projeto, por outro, este deve também respeitar o modo de funcionamento da **EF**(Entity Framework 6), a ferramenta de comunicação entre a aplicação construída em **.Net 5.0** e o motor de base de dados escolhido, o **Microsoft SQL Server**.

No que diz respeito aos requisitos, estes indicam quais as entidades que caracterizam o projeto, detalhando também os atributos de cada uma e a forma como se relacionam entre si.

A **EF**, quando conjugada com a framework web a utilizar, o **ASP.NET MVC**, utiliza uma **code-first approach**, isto é, uma metodologia em que as várias entidades devem ser definidas através de código **C#**, sendo os seus atributos declarados sobre a forma dos tipos de dados existentes na linguagem. A **EF** irá, depois, ser responsável por migrar as entidades declaradas utilizando código C para o sistema de gestão de base de dados a utilizar, o **SQL Server**, criando as tabelas, atributos, índices e gerindo as relações entre as várias tabelas. Este mapeamento segue regras próprias de conversão da ferramenta que devem ser tomadas em conta no planeamento da base de dados.

O caso mais notável de fatores específicos da stack tecnológica a ter em conta relaciona-se com os utilizadores. Toda a lógica de criação, autenticação e autorização de utilizadores em asp.net utiliza modelos e entidades com atributos pré-definidos, criando tabelas correspondentes não a modelos planeadas pelo programador mas sim derivadas dos modelos pré-definidos da lógica de utilizadores do asp.net. Os detalhes das entidades relativas a autenticação e autorização de utilizadores serão discutidos numa secção posterior do relatório.

13.1 Identificação e Caracterização das Entidades a partir de Requisitos

O primeiro vetor a ter em conta no planeamento do sistema de base de dados são os requisitos da aplicação levantados. Atendendo a estes foi identificada a necessidade de criação das seguintes entidades:

- **Câmara**

A entidade Câmara representa cada uma das câmaras de deteção espalhadas pelo Parque Nacional da Peneda-Gerês. Esta será caracterizada pelo seu **Id**, **Nome**, **Latitude**, **Longitude**, se se encontra numa **zona de acesso restrito** e o seu **acesso às suas deteções** é restrito a administradores apenas ou ao público geral.

- **Espécie**

A entidade Câmara representa cada das espécies que poderão ser detetadas pelas câmaras do Parque Nacional da Peneda-Gerês. Esta será caracterizada pelo seu **Id**, **Nome Comum**, **Nome científico**, **Descrição**, se tem **comportamentos predatórios** e ainda o url da **imagem** associada.

A entidade relacionada com os utilizadores não foi aqui descrita visto dependente da framework utilizada, sendo descrita na próxima secção.

13.2 Identificação e Caracterização das Entidades derivadas pela framework Web

Uma das entidades definidas pelos requisitos levantados é o **utilizador**. Infelizmente, a caracterização desta entidade não se pode limitar à identificação dos atributos relevantes à aplicação, este facto deve-se à forma como a framework **ASP.NET** lida com a lógica de utilizadores.

Quando se pretende que uma aplicação construída em ASP.NET suporte contas individuais, deve-se incluir a componente **Individual Identity**.

Esta componente introduz modelos responsáveis pela lógica tanto de autenticação como de autorização, isto é, o registo de utilizadores, o login de utilizadores e a atribuição de permissões (como, por exemplo, permissões de administrador).

As alterações introduzidas por esta componente são as seguintes: 6 novas entidades, 4 novos relacionamentos 1:N e um relacionamento N:M. Este número extenso de entidades, deve-se à capacidade do módulo suportar várias metodologias de autenticação (como usar *login providers* externos), e de autorização, muitas delas análogas a redes sociais, o que, dado o facto da aplicação a desenvolver não se encontrar nessa categoria, as deixará inutilizadas. Porém, concordou-se que seria necessário identificá-las como entidades para dar uma ideia da verdadeira estrutura da base de dados, em caso de posterior expansão das funcionalidades do projeto.

Explicitam-se, de seguida, as entidades criadas relevantes ao projeto:

- **AspNetUsers**

Esta é a entidade central da lógica de autenticação, representando um utilizador. Por base um utilizador é caracterizado pelo seu **Id**, **username**, **username normalizado**, **email**, **email normalizado**, **se o email foi confirmado**, a **password encriptada**, uma **estampa de segurança**, uma **estampa de controlo de concorrência** no acesso à base de dados, **número de telemóvel**, um **indicativo se o número foi confirmado**, se o utilizador tem **autenticação de 2 fatores** ativada, se o utilizador está **banido**, **o tempo em que a sua proibição termina** e o número de **tentativas de acesso falhadas**.

À entidade base criada pela framework, foram ainda adicionados os seguintes campos extraídos dos requisitos associados a utilizadores: um **indicativo se pretende receber o resumo diário** e, em caso afirmativo, **se pretende receber por telemóvel e/ou por email**.

- **AspNetRoles**

Esta é a entidade central da lógica de autorização. Representa os *roles* ou **níveis de autorização** existentes na aplicação, não representando assim as autorizações de cada utilizador mas sim as várias possibilidades de autorização possível. Cada role é caracterizado pelo seu **Id**, **nome normalizado** e uma **estampa de concorrência**

O módulo *Identity* do ASP.NET suporta 2 mecanismos diferentes de autorização: **role-based** e **claims-based**. O projeto utilizará autorização **role-based**, pois esta é suficiente em ambientes cujos níveis de acesso estão bem definidos. No entanto estas serão definidos de seguida para propósitos de esclarecimento.

A mesma filosofia aplica-se a outra funcionalidade não utilizada: a de utilizador utilizar login providers externos. Esta não será utilizada no projeto, porém o módulo de identidade automaticamente cria as entidades, pelo que irão ser descritas.

- **AspNetUserClaims**

Esta entidade representa uma *claim* pertencente a um utilizador. É caracterizada pelo seu **Id**, **tipo de claim** e **valor da claim**.

- **AspNetRoleClaims**

Esta entidade permite juntar autorização role-based com claim-based, juntando claims a roles. É caracterizada pelo seu **Id**, **tipo** e **valor**.

- **AspNetUserLogins**

Esta entidade representa o login de um utilizador utilizando *providers* externos. É caracterizada pelo **id do provider**, **chave do provider** e o **nome do utilizador** no provider.

- **AspNetUserTokens**

Esta entidade representa os tokens atribuídos ao utilizador quando usa *login providers* externos. É caracterizada pelo **Id do utilizador** (que serve tanto de chave primária como estrangeira, pelo que deve ser mencionada), **Id do login provider**, **nome** do token e **valor**.

Por fim, deve salientar que a **Entity Framework 6**, funciona através de "migrations". Estas não correspondem ao conceito habitual de mover dados de uma base de dados para outra, mas sim de alterações à estrutura da base de dados atual, como adicionar novas tabelas e mudar o nome de certas colunas. De forma a manter um histórico destas migrações, de forma a poder reverter mudanças indesejadas, a EF guarda informação sobre estas numa tabela na própria base de dados

Esta tabela tem nome de **_EFMigrationsHistory** e é caracterizada pelo seu **Id** e pela **versão da migração**.

13.3 Identificação e Caracterização dos Relacionamentos

- **Avistamentos**

Relacionamento: Avistamento liga a câmara que detetou a espécie num específico momento a essa espécie.

Descrição: As várias câmaras espalhadas pelo parque vão detetando espécies que passam nas zonas onde estão posicionadas.

Cardinalidade: Câmara (0..*) - Espécie (0..*).

Cada câmara captura avistamentos cada um contendo uma espécie, sendo possível capturar a mesma espécie em momentos diferentes, tendo vários avistamentos. Em sentido inverso uma espécie pode ser detetada várias vezes por várias câmaras.

Atributos: Este relacionamento é caracterizado pelo seu **Id**, o **Id da câmara** que detetou a espécie, o **Id da espécie**, o **número de indivíduos detetados** e ainda o

momento de deteção

▪ **Espécie Seguida**

Relacionamento: Utilizador segue várias espécies

Descrição: Cada utilizador escolhe receber notificações de avistamentos de um conjunto de espécies

Cardinalidade: (ASPNetUsers) Utilizador (1..*) – Espécie (1..*).

Cada utilizador segue um conjunto de espécies. Cada espécie é seguida por um conjunto de utilizadores.

Atributos: Este relacionamento não possui atributos.

▪ **Câmara Seguida**

Relacionamento: Utilizador pretende ver várias câmaras.

Descrição: Cada utilizador escolhe receber notificações de avistamentos capturados por várias câmaras

Cardinalidade: (ASPNetUsers) Utilizador (1..*) – Câmara (1..*).

Cada utilizador segue um conjunto de câmaras. Cada câmara é seguida por um conjunto de utilizadores.

Atributos: Este relacionamento não possui atributos.

▪ **ASPNetUserRoles**

Relacionamento: Utilizador tem várias *roles*.

Descrição: Cada utilizador pode ter um ou mais *roles*

Cardinalidade: (ASPNetUsers) Utilizador (1..*) – (ASPNetRoles) *Role* (0..*).

Cada utilizador pode ter 0 ou mais *roles*. Cada *role* pode ter 0 ou mais utilizadores.

Atributos: Este relacionamento não possui atributos.

Relacionamentos entre tabelas não utilizadas (claims e login providers) são omitidos desta secção de descrição, sendo apenas apresentados no modelo lógico.

13.4 Dicionário de dados

A tabela 13.1 apresenta o dicionário de dados, onde, cada entidade é descrita segundo os seus atributos. Por motivos de simplificação e eliminação de duas colunas, um atributo que seja uma chave primária é assinalada a verde e uma chave estrangeira é colorida a laranja. Visto este dicionário caracterizar as entidades quando estas já estarão implementadas na base de dados, são omitidas as colunas: composto, multivalor, derivado e candidato.

Entidade	Atributos	Tipo de Dados	Refere	Opcional
Camera	Id	int	–	Não
	Name	nvarchar(100)	–	Não
	Latitude	real	–	Não
	Longitude	real	–	Não
	RestrictedZone	bit	–	Não
	RestrictedArea	bit	–	Não
Species	Id	int	–	Não
	CommonName	nvarchar(100)	–	Não
	ScientificName	nvarchar(100)	–	Não
	Description	nvarchar(max)	–	Sim
	IsPredatory	bit	–	Não
	Image	nvarchar(100)	–	Não
Sighting	Id	int	–	Não
	CameraId	int	Camera	Não
	SpeciesId	int	Species	Não
	CaptureMoment	datetime2	–	Não
	Quantity	int	–	Não
FollowedCamera	Id	int	–	Não
	CameraId	int	Camera	Não
	UserId	nvarchar(450)	AspNtUsers	Não
FollowedSpecies	Id	int	–	Não
	SpeciesId	int	Species	Não
	UserId	nvarchar(450)	AspNetUsers	Não
AspNetUsers	Id	nvarchar(450)	–	Não
	ReceiveSummary	bit	–	Não
	UseCellhphone	bit	–	Não
	UseEmail	bit	–	Não
	Username	bit	–	Sim
	NormalizedUseEmail	bit	–	Sim
	EmailConfirmed	bit	–	Não
	PasswordHash	nvarchar(max)	–	Sim
	SecutiryStamp	nvarchar(max)	–	Sim
	ConcurrencyStamp	nvarchar(max)	–	Sim
	PhoneNumber	nvarchar(max)	–	Sim

Entidade	Atributos	Tipo de Dados	Refere	Opcional
	PhoneNumberConfirmed	bit	–	Não
	TwoFactorEnabled	bit	–	Não
	LockoutEnd	datetimeoffset	–	Sim
	LockoutEnabled	bit	–	Não
	AccessFailedCount	bit	–	Sim
AspNetRoles	Id	nvarchar(450)	–	Não
	Name	varchar(256)	–	Sim
	NormalizedName	nvarchar(256)	–	Sim
	ConcurrencyStamo	nvarchar(max)	–	Sim
AspNetUserRoles	UserId	nvarchar(450)	AspNetUsers	Não
	RoleId	nvarchar(450)	AspNetRoles	Não
_EFMigrationsHistory	MigrationId	nvarchar(150)	–	Não
	ProductVersion	nvarchar(32)	–	Não

Tabela 13.1: Dicionário de dados

O dicionário de dados presente na tabela 13.2 detalham as entidades que não serão utilizadas no projeto, no entanto são criadas pelo módulo *Identity* e, como tal, devem ser documentadas de forma a viabilizar o atualizações futuras ao projeto.

Entidade	Atributos	Tipo de Dados	Refere	Opcional
AspNetUserLogins	LoginProvider	nvarchar(128)	–	Não
	ProviderKey	nvarchar(128)	–	Não
	UserId	nvarchar(450)	AspNetUsers	Não
	ProviderDisplayName	nvarchar(max)	–	Sim
AspNetUserClaims	Id	int	–	Não
	UserId	nvarchar(450)	AspNetUsers	Não
	ClaimType	nvarchar(max)	–	Sim
	ClaimValue	nvarchar(max)	–	Sim
AspNetUserTokens	UserId	nvarchar(450)	AspNetUsers	Não
	LoginProvider	nvarchar(128)	–	Não
	Name	nvarchar(128)	–	Não
	Value	nvarchar(max)	AspNetUsers	Sim
AspNetRoleClaims	Id	int	–	Não
	RoleId	nvarchar(450)	AspNetRoles	Não
	ClaimType	nvarchar(max)	–	Sim
	ClaimValue	nvarchar(max)	–	Sim

Tabela 13.2: Dicionário de dados de entidades não utilizadas

13.5 Domínio dos atributos

O seguinte conjunto de tabelas descreve os domínios dos vários atributos das entidades que serão utilizadas no projeto, isto é, as presentes na tabela 13.1. Chaves estrangeiras são ocultadas pois o seu domínio é derivado da chave primária da entidade que referem e, como tal, entidades que representem relações e não têm outros atributos serão omitidas.

▪ Camera

Atributo	Tipo de Dados	Domínio de valores
Id	int	Número inteiro positivo.
Name	nvarchar(100)	Sequência de palavras
Latitude	real	Valor dentro dos limites do parque (41.33 a 41.93)*
Longitude	real	Valor dentro dos limites do parque (-8.12 a -8.06)*
RestrictedZone	bit	0/1 (Livre/Restrita)
RestrictedArea	bit	0/1 (Livre/Restrita)

Tabela 13.3: Caracterização dos atributos de Camera

* Os limites do Parque Nacional Peneda-Gerês não são retangulares, pelo que nem sempre uma câmara nesta latitude/longitude é válida, o intervalo apresentado apenas apresenta o valor mínimo e máximo que poderá tomar nos extremos do parque.

▪ Species

Atributo	Tipo de Dados	Domínio de valores
Id	int	Número inteiro positivo.
CommonName	nvarchar(100)	Sequência de palavras
ScientificName	nvarchar(100)	Sequência de palavras
Description	nvarchar(max)	Sequência de palavras
IsPredatory	bit	0/1 (Não/Sim)
Image	nvarchar(100)	nome de ficheiro contendo também extensão

Tabela 13.4: Caracterização dos atributos de Species

▪ Sighting

Atributo	Tipo de Dados	Domínio de valores
Id	int	Número inteiro positivo.
CaptureMoment	datetime2	Formato AAAA-MM-DD hh:mm
Quantity	int	Número inteiro positivo

Tabela 13.5: Caracterização dos atributos de Sighting

▪ Utilizador (ASPNetUsers)

Atributo	Tipo de Dados	Domínio de valores
Id	nvarchar(450)	Identificador composto por sequência de caracteres.
ReceiveSummary	bit	0/1 (Sim/Não)
UseCellphone	bit	0/1 (Sim/Não)
UserName	nvarchar(256)	Sequência de caracteres
NormalizedUserName	nvarchar(256)	Sequência de caracteres ASCII em maiúscula e sem espaços
Email	nvarchar(256)	Sequência de caracteres que seguem o formato de um email
NormalizedEmail	nvarchar(256)	Sequência de caracteres ASCII em maiúscula e sem espaços que seguem o formato de um email
EmailConfirmed	bit	0/1(Não/Sim)
PasswordHash	nvarchar(max)	sequência de caracteres que representam o output de uma função de hashing
SecurityStamp	nvarchar(max)	sequência de caracteres gerada pelo sistema
ConcurrencyStamp	nvarchar(max)	sequência de caracteres gerada pelo sistema
PhoneNumber	nvarchar(max)	número inteiro positivo que represente um número móvel
PhoneNumberConfirmed	bit	0/1(Não/Sim)
TwoFactorEnabled	bit	0/1(Não/Sim)
LockoutEnd	datetimeoffset	Formato AAAA-MM-DD hh:mm +/- hh:mm
LockoutEnabled	bit	0/1(Não/Sim)
AccessFailedCount	int	Número inteiro positivo

Tabela 13.6: Caracterização dos atributos de Utilizador (AspNetUsers)

▪ AspNetRoles

Atributo	Tipo de Dados	Domínio de valores
Id	nvarchar(450)	Identificador composto por sequência de caracteres.
Name	nvarchar(256)	Sequência de caracteres
NormalizedName	nvarchar(256)	Sequência de caracteres ASCII em maiúscula e sem espaços
ConcurrencyStamp	nvarchar(max)	sequência de caracteres gerada pelo sistema

Tabela 13.7: Caracterização dos atributos de AspNetRoles

- **_EFMigrationsHistory**

Atributo	Tipo de Dados	Domínio de valores
MigrationId	nvarchar(150)	Identificador composto pela estampa com a data de criação da migração e o nome da migração .
ProductVersion	nvarchar(32)	Versão da Entity Framework

Tabela 13.8: Caracterização dos atributos de _EFMigrationHistory

13.6 Modelo lógico

A figura 13.1 apresenta o modelo lógico da base de dados do projeto, projetada para o motor de base de dados **Microsoft SQL Server**. Este apresenta as várias tabelas existentes na base de dados assim como os seus atributos, como detalhados nas secções anteriores.

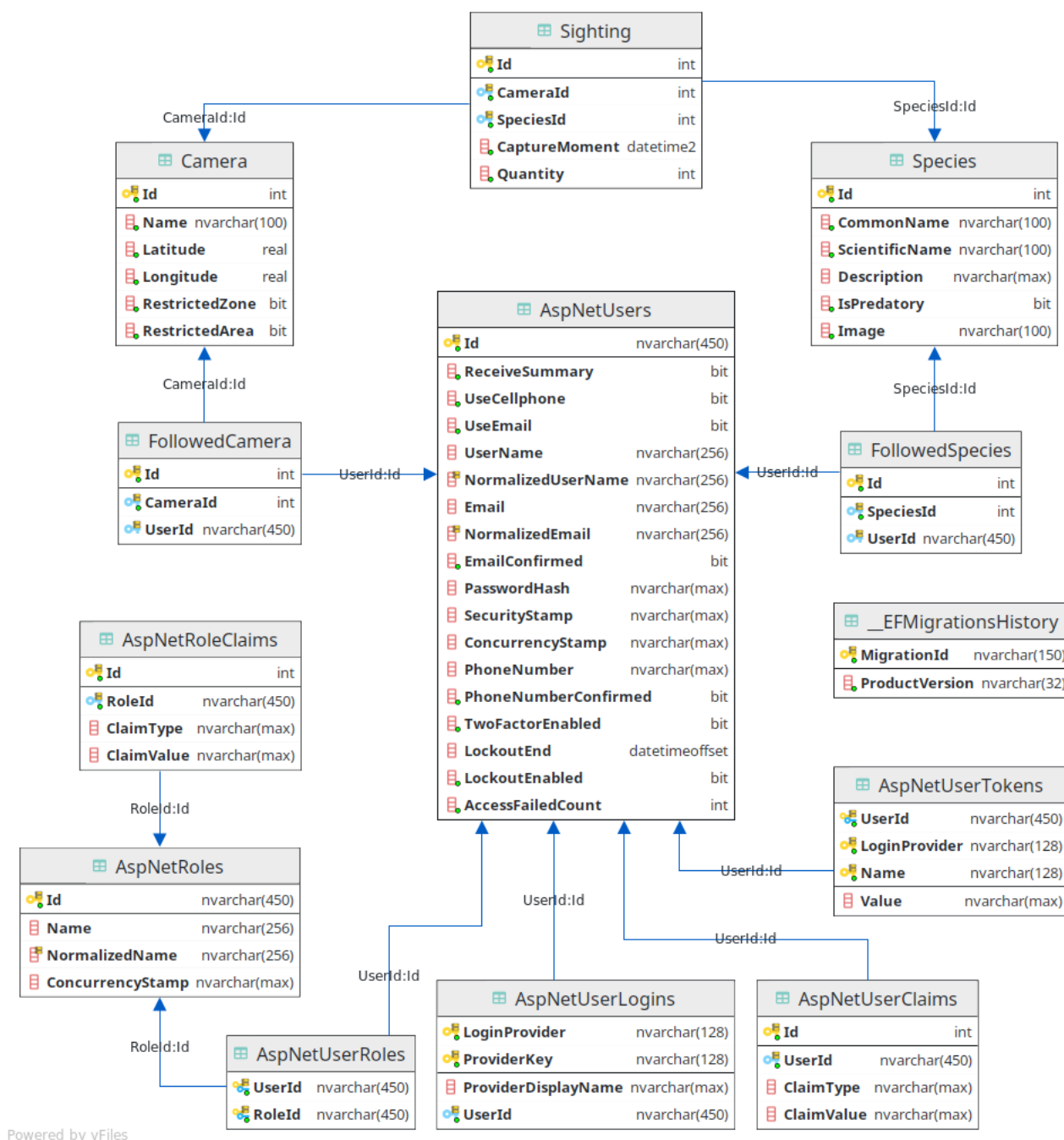


Figura 13.1: Modelo lógico da base de dados.

13.7 Estimativa do Espaço em Disco da Base de Dados e Taxa de Crescimento Anual

De forma a estimar o espaço gasto por uma entrada de cada entidade existente na nossa Base de Dados foi necessário calcular quantos bytes serão gastos por cada tipo de dados. Para tal, foi consultada a documentação do Microsoft SQL Server

Dessa consulta, estabeleceram-se as seguintes métricas de bytes por tipo de dado:

- **int**: 4 bytes;
- **real**: 4 bytes;
- **datetime2**: 6 bytes;
- **nvarchar(X)**: $2 \cdot X + 2$ bytes;
- **bit**: 1 byte (escolha do sql server);
- **datetimeoffset**: 10 bytes.

Assim sendo, procede-se ao cálculo do tamanho gasto por uma entrada de cada entidade, tendo por base a descrição dos tipos de dados de cada atributo, apresentados nas secções anteriores.

N.B.: Embora o tipo de dados nvarchar seja de tamanho variável, assume-se, para propósitos de uma previsão cuidadosa, que todos os caracteres são gastos. Quando se trata de nvarchar(max) é feita uma estimativa do tamanho médio desse campo.

N.B. 2: Embora esteja marcado como nvarchar(max), o comprimento de uma securityStamp é sempre de 32 caracteres, o de uma concurrencyStamp 36 caracteres e o de uma passwordHash 84.

- **Camera**: $4 + (2 \cdot 100 + 2) + 4 + 4 + 1 + 1 = \mathbf{216 \text{ bytes}}$;
- **Species**: $4 + (2 \cdot 100 + 2) + (2 \cdot 100 + 2) + (2 \cdot 200 + 2) + 1 + (2 \cdot 100 + 2) = \mathbf{1010 \text{ bytes}}$;
- **Sighting**: $4 + 4 + 4 + 6 + 4 = \mathbf{22 \text{ bytes}}$;
- **FollowedSpecies**: $4 + 4 + (2 \cdot 450 + 2) = \mathbf{910 \text{ bytes}}$;
- **FollowedCamera**: $4 + 4 + (2 \cdot 450 + 2) = \mathbf{910 \text{ bytes}}$;
- **ASPNetUsers**: $(2 \cdot 450 + 2) + 1 + 1 + 1 + (2 \cdot 256 + 2) + (2 \cdot 256 + 2) + (2 \cdot 256 + 2) + (2 \cdot 256 + 2) + 1 + (84 \cdot 2 + 2) + (32 \cdot 2 + 2) + (36 \cdot 2 + 2) + 1 + 1 + 10 + 1 + 4 = \mathbf{3289 \text{ bytes}}$;

- **AspNetRoles:** $(2*450+2) + (2*256+2) + (2*256+2) + (36*2+2) = 2514$ bytes;
- **AspNetUserRoles :** $(2*450+2) + (2*450+2) = 1804$ bytes.

Nestes cálculos não foram incluídos cálculos relativos ao tamanho dos índices, devido à complexidade do cálculo do custo destes no SQL Server. Estes estão presentes em todas as chaves primárias e estrangeiras.

13.7.1 Previsão do tamanho inicial

Assuma-se, por base, que, inicialmente, a base de dados estará preenchida com os seguintes dados:

- 20 câmaras;
- 20 espécies;
- 10 utilizadores;
- Cada utilizador segue as 20 espécies;
- Cada utilizador segue as 20 câmaras;
- 2 *roles*;
- Todos os utilizadores têm o *role* admin e um o *role* root;
- 500 avistamentos.

Tendo em conta os pressupostos anteriores, a base de dados ocupará, inicialmente **457.3 Kilobytes**.

13.7.2 Previsão da taxa de crescimento anual

Assumem-se, agora, os seguintes pressupostos de crescimento anual:

- 1 novo avistamento a cada 5 minutos => 105120 avisamentos por ano;
- 100 novos utilizadores por mês => 1200 utilizadores por ano;
- Cada utilizador segue todas as 20 espécies e 20 câmaras;
- Nenhum destes utilizadores têm um *role* (pois este é reservado a administradores).

Como tal, verifica-se que o aumento de tamanho da base de dados, por ano, será de **49**

Megabytes por ano. Reconhece-se que este valor poderá chegar a perto do dobro devido à existência de índices na base de dados.

14 Definição e caracterização do sistema de interface (mockups)

Após a realização do levantamento e análise de requisitos, de um considerável número de diagramas e de uma melhor caracterização da base de dados do projeto estamos agora prontos para poder utilizar o conhecimento adquirido para projetar alguns *mockups* de como será a interface de comunicação do utilizador com o sistema.

Ao longo deste capítulo serão utilizadas algumas imagens desenvolvidas pelo grupo com o intuito acima descrito, além disso, estas imagens serão acompanhadas de uma breve descrição da forma de funcionamento dos pontos principais nela apresentados.

A página principal do sistema será a página responsável pela autenticação e registo do utilizador, sendo essa página bastante comum o nosso grupo decidiu não fazer um mockup da mesma de modo a poder focar-se mais na interface pós autenticação sendo esta a mais pertinente.

14.1 Dashboard

Após o utilizador efetuar o registo/autenticação na aplicação será apresentado com uma interface que se assemelhará à interface presente na figura 14.1.

O cabeçalho apresentado no topo da página irá conter o nome da aplicação seguido do Nome do Utilizador atualmente autenticado, neste caso, André Belo. Além disso teremos também no cabeçalho hiperligações para outras 2 páginas que iremos explicar futuramente denominadas **Câmaras** e **Espécies**.

Após isso temos um mapa obtido através da API do Bing Maps, além disso podemos ver que o mapa em questão tem nele presente vários marcadores assinalados a verde que identificam algumas câmaras espalhadas ao longo da Serra Peneda-Gerês, estes marcadores são obtidos através de uma pesquisa na tabela **"Camera"** cruzada com a tabela **"FollowedCamera"** usando o UerId do Utilizador atualmente autenticado. Vemos também que existe a possibilidade de aumentar a aproximação e diminuir a mesma de modo a poder ter uma melhor visão da localização das câmaras em questão caso estejam muito próximas.

Cada um dos marcadores assinalados podem ser clicados apresentando alguma da meta-informação da câmara, sendo depois possível clicar no nome da câmara para ser redirecionado

para a página **Câmaras**, onde serão apresentados os vários detalhes dessa câmara.

Por último vemos que existe uma aba intitulada **Notificações de Avistamentos**. Esta aba servirá para, como o nome indica, apresentar as notificações que o Utilizador definir que deseja receber, estas notificações, como referido anteriormente, podem ser referentes a quais câmaras/espécies deseja ser notificado. Para a implementação desta funcionalidade, sempre que um avistamento for recebido pela aplicação, caso esse avistamento confira os requisitos das preferências definidas pelo utilizador será apresentado na sua sessão de browser.

Cada uma das espécies presente na aba de notificações pode também ser clicada de modo a redirecionar o utilizador para a página **Espécies** onde serão apresentados mais detalhes sobre a espécie em questão, assim como, os últimos avistamentos da mesma.

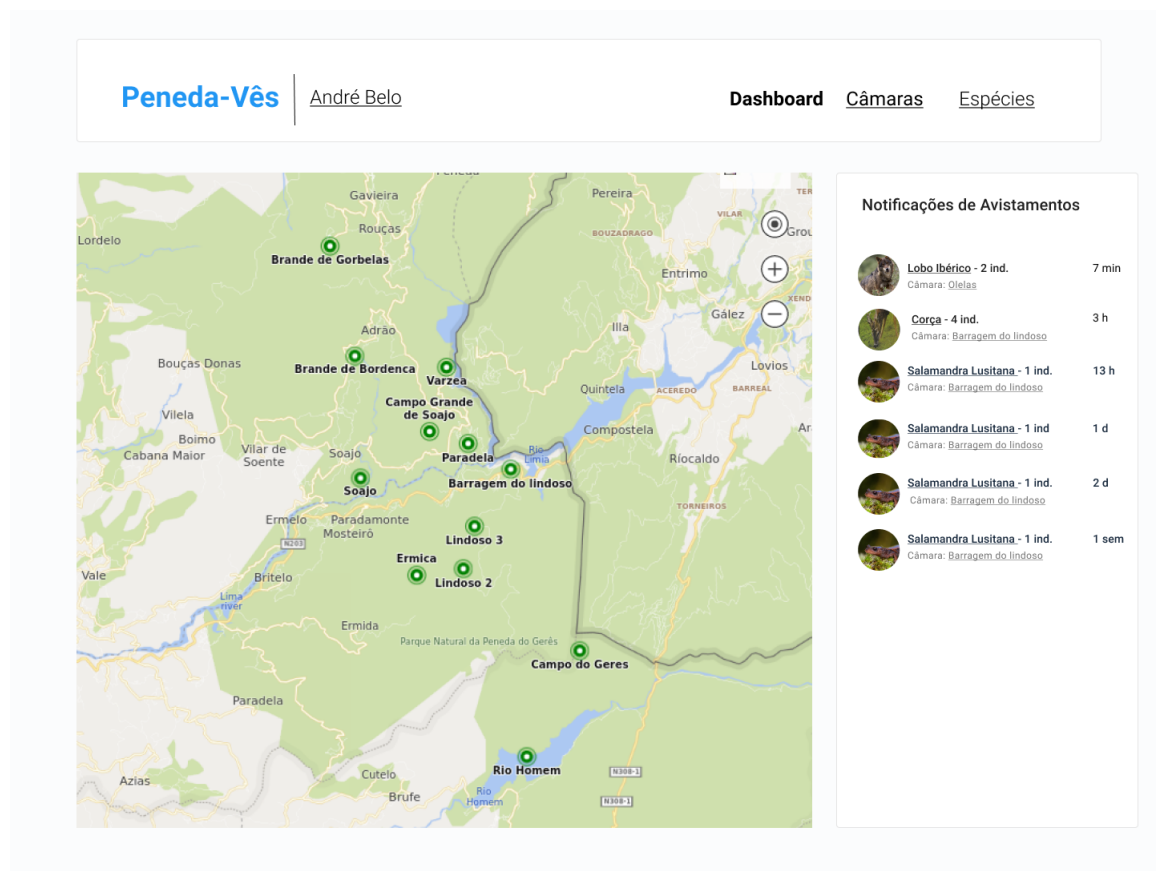


Figura 14.1: Mockup do Dashboard

14.2 Página de uma câmara

Esta página será responsável por apresentar ao Utilizador os detalhes de uma câmara escolhida na página **Dashboard** ou na lista apresentada na página **Câmaras**. É possível obter uma melhor visão da explicação dada com o auxílio da figura 14.2.

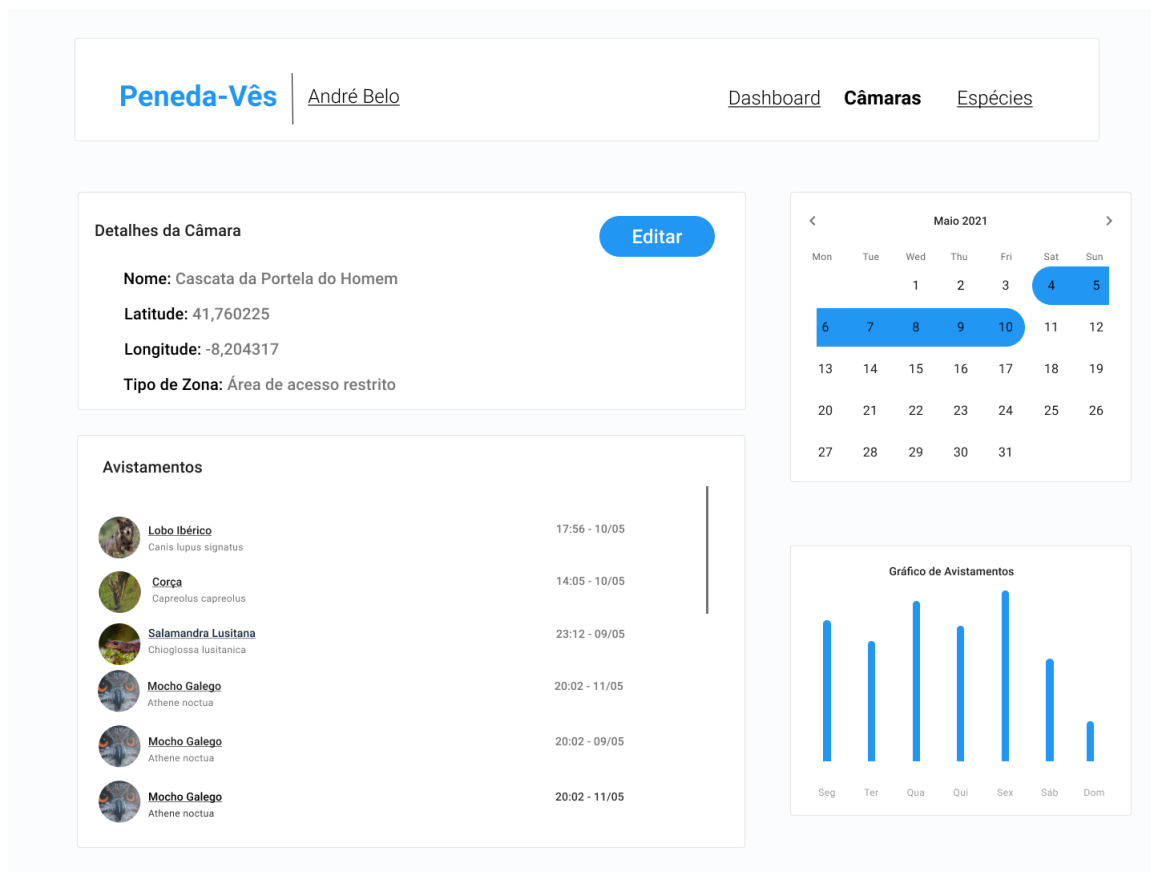


Figura 14.2: Mockup da página de uma câmara

Sendo o cabeçalho no topo da página algo que aparece em todas as páginas e tendo sido referida e descrita na secção anterior não será referida nesta secção.

Nesta página são nos apresentadas 4 secções diferentes.

A primeira secção, denominada **Detalhes da Câmara**, é responsável por apresentar alguma meta informação da câmara, com a opção **Editar** caso o Utilizador autenticado seja o administrador, estas informações incluem Nome atribuído à câmara no momento de criação da mesma, a posição (latitude e longitude) e o tipo de zona da mesma, indicando se essa zona é ou não uma área restrita. Estas informações são facilmente obtidas através de uma pesquisa na tabela **"Camera"** pelo Id da câmara obtendo os valores das outras colunas.

A segunda secção é um calendário onde o Utilizador pode escolher o intervalo de tempo no qual deseja ver os Avistamentos feitos pela Câmara em questão, esta seleção de avistamentos é apresentada na secção de **Avistamentos**.

A terceira secção, denominada **Avistamentos**, será um conjunto de Avistamentos feitos pela Câmara no espaço de tempo determinado, estes avistamentos têm informação básica sobre a espécie avistada, caso o Utilizador deseje obter mais informação sobre a mesma pode clicar no

nome dela que contém uma hiperligação para a página dessa espécie onde será possível obter mais informação sobre a espécie em questão.

A quarta e última secção da página, denominada **Gráfico de Avistamentos**, prende-se com uma apresentação do gráfico de avistamentos registados pela câmara no espaço de tempo determinado tempo, este gráfico será apresentado de forma a que o Utilizador possa ter uma melhor noção dos dias em que mais avistamentos foram feitos e a evolução dos mesmos ao longo do tempo. Este gráfico será construído através dos vários avistamentos obtidos na tabela "**Sighting**" cruzada com a tabela "**FollowedCamera**" utilizando o UserId do Utilizador atualmente autenticado.

14.3 Página de uma espécie

Esta página será responsável por apresentar ao Utilizador os detalhes de uma espécie escolhida na página "**Dashboard**" ou no menu apresentado na página "**Espécies**". É possível obter uma melhor visão da explicação dada com o auxílio da figura 14.3.

Como já foi referido na secção anterior a fita no topo da página esta presente em todas as páginas e já foi descrita na primeira secção deste capítulo não sendo assim repetida a informação já dada.

Está página assemelha-se à página anterior na medida em que são apresentadas 4 secções diferentes e têm todas uma certa semelhança às páginas anteriormente descritas.

A primeira secção desta página, denominada **Detalhes da Espécie**, é responsável por apresentar a espécie em questão, identificando-o pelo seu nome comum, nome científico e o carácter da mesma. Após isto temos uma breve descrição da espécie assim como uma imagem de forma a que os utilizadores possam visualizar a espécie em questão. Para os Utilizadores com poder de administração é apresentada uma opção para editar as informações da espécie em questão. Estas informações são facilmente obtidas através de uma pesquisa na tabela "**Species**" pelo Id da espécie obtendo os valores das outras colunas.

A segunda secção desta página assemelha-se em grande parte à segunda secção da página **Câmaras**, esta secção irá permitir ao utilizador escolher os dias em que pretende ver listados os avistamentos da espécie, esta listagem é feita na secção **Avistamentos**.

A terceira secção desta página, denominada **Avistamentos**, apresenta nela listadas todas as câmaras que fizeram avistamentos da espécie em questão, estes avistamentos são feitos tendo em conta o espaço de tempo escolhido pelo utilizador na secção anteriormente falada.

A quarta e última secção desta página, denominada **Gráfico de Avistamentos**, prende-se com a apresentação ao Utilizador dos vários Avistamentos da espécie em questão agrupados por dias, este espaço de tempo é definido na segunda secção da página de modo a dar mais controlo ao Utilizador sobre os dados apresentados. Este gráfico será construído através dos

vários avistamentos obtidos na tabela **"Sighting"** cruzada com a tabela **"FollowSpecies"** utilizando o UserId do Utilizador atualmente autenticado e apresentado ao Utilizador.

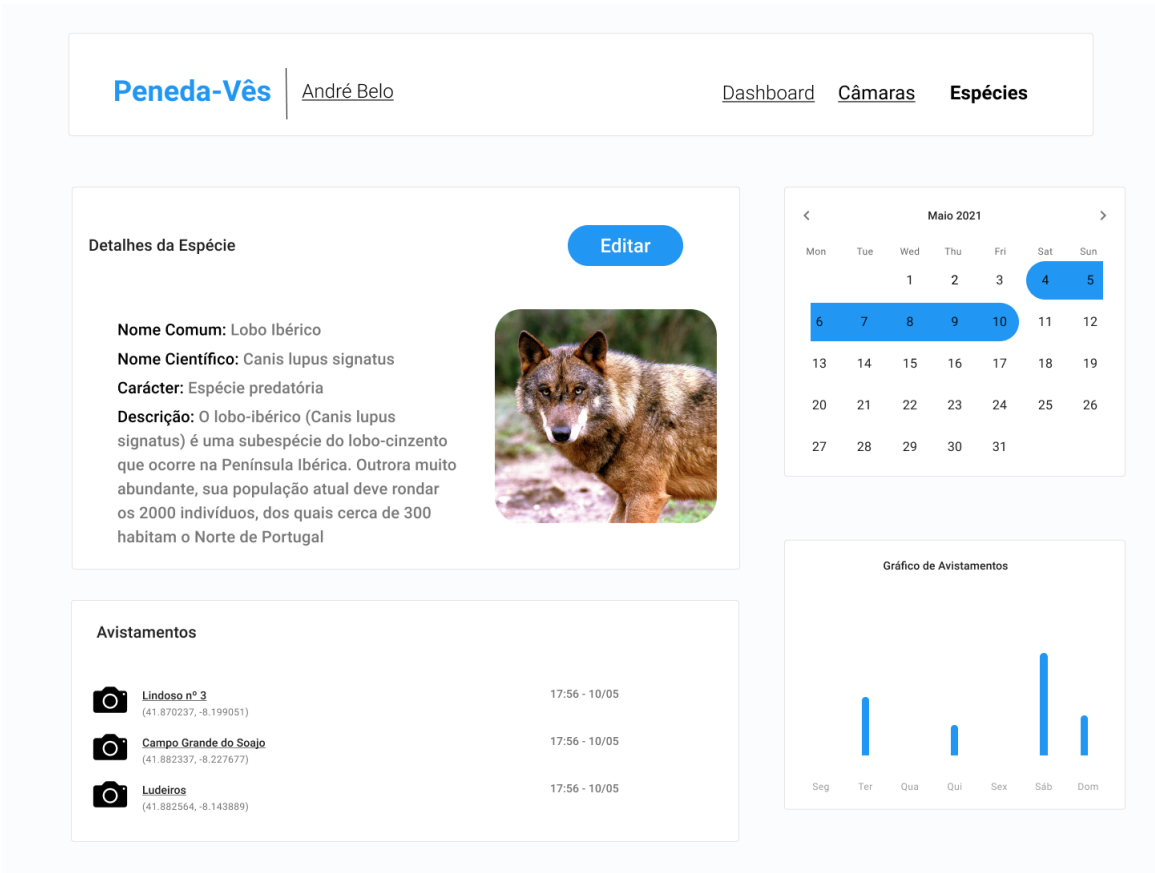


Figura 14.3: Mockup da página Espécies

15 Conclusões e Trabalho Futuro

Ao longo de ambas as fases do projecto, planeou-se o desenvolvimento de um software idealizado pela equipa: uma aplicação de rastreio e monitorização de animais, através de câmaras instaladas no Parque Nacional Peneda-Gerês. Este teve especial atenção em garantir que existe um planeamento fiável e bem estruturado, de modo a que desde uma fase inicial se tivesse uma perspetiva das especificidades do meio em que a aplicação vai ser utilizada. Deste modo, assegura-se o cumprimento de todos os seus objetivos, e garante-se assim a utilidade que é desejada.

Para o efeito, foram estabelecidas as funcionalidades pretendidas, obtendo-se um bom planeamento da aplicação a implementar, adequando-a ao que será exequível pela equipa no tempo atribuído para o projeto.

Posto isto, acredita-se que se tem uma estrutura consistente, e que permitirá, de modo iterativo, dar início à fase da implementação.

Vai ser dada especial atenção ao programa que simulará os dados referentes às movimentações dos animais do Parque, uma vez que a credibilidade do seu algoritmo de simulação será crucial para que a aplicação gere dados fidedignos e que poderão ser utilizados para uma aplicação do sistema num contexto real.

Em relação à primeira fase, foi atualizado o diagrama de gantt de forma a refletir realmente o comportamento adotado e tempo despendido em cada tarefa na segunda fase.

Em conclusão, crê-se que foram abordados todos os pontos necessários a uma estruturação calendarizada e metódica, de forma a que o projeto esteja concluído nas datas previstas, e cumprindo com o orçamento especificado.

Bibliografia

- [1] Just Natur, 2017. *Fauna do Parque Nacional Peneda Gerês*. [online] Disponível em: <<http://www.justnatur.com/fauna-do-Parque-Nacional-peneda-geres>> [Acedida a 19 de Março de 2021].
- [2] Serra do Gerês, 2018. *Fauna*. [online] Disponível: <<https://www.serradogeres.com/index.php/natureza/fauna>> [Acedida a 19 de Março de 2021].
- [3] Microsoft, 2021. *ASP.NET documentation*. Disponível em: <<https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-5.0>> [Acedida a 28 de Março de 2021]
- [4] Microsoft, 2021. *SQL Server data types*. Disponível em: <<https://docs.microsoft.com/en-us/sql/t-sql/data-types/data-types-transact-sql?view=sql-server-ver15>> [Acedida a 25 de Abril de 2021]

Lista de Siglas e Acrónimos

UML Unified Modelling Language

ASP Active Server Pages

SQL Structured Query Language

API Application Programming Interface

SMS Short Message Service

MVC Model-View-Controller

HTTP HyperText Transfer Protocol

HTML HyperText Markup Language

CSS Cascading Style Sheets

Anexos

Use Case		
Use Case:	Utilizar dashboard principal	
Ator:	Utilizador Base	
Descrição	Permite a um utilizador observar os avistamentos registados no parque pelas câmaras e espécies por ele pedidas	
Pré-condição:	Utilizador está autenticado	
Pós-condição:	Avistamentos são apresentados ao utilizador	
	Ator	Sistema
	1. Utilizador visita página do dashboard principal	
		2. Sistema seleciona, da base de dados, as câmaras que o utilizador indicou, nas definições, que pretende seguir
		3. Por cada uma dessas câmaras, o sistema seleciona, da base de dados, os avistamentos de espécies que o utilizador pretende seguir
		4. Sistema chama a API do Bing Maps de forma a obter o mapa do parque nacional da peneda-gerês.
		5. Sistema projeta, sobre o mapa, cada uma das câmaras, utilizando a informação da latitude e longitude nelas contida
		6. Sistema adiciona informação a cada uma das câmaras projetadas no mapa, nomeadamente o seu nome, o número de avistamentos detetados, num intervalo pré-definido, e associa à imagem da câmara uma hiperligação clicável que permite explorar a câmara, onde está incluída a sua lista de avistamentos completa.
		7. Sistema projeta também informação dos vários avistamentos, ordenados de forma cronológica decedente, numa zona da vista separada da do mapa.
		8. Sistema devolve a vista ao utilizador

Figura 15.1: Utilizar Dashboard Principal

Use Case		
Use Case:	Consultar Espécies	
Ator:	Utilizador Base	
Descrição:	Permite ao utilizador consultar as espécies existentes no parque, assim como os detalhes a elas associados	
Pré-condição:	Gestor está autenticado	
Pós-condição:	Gestor recebe listagem das espécies e detalhes associados	
	Ator	Sistema
Cenário Normal	1. Utilizador solicita a listagem das espécies na aplicação	
		2. Sistema seleciona, da base de dados as espécies existentes no parque, apresentando-as ao utilizador
	3. Utilizador clica numa espécie	
		4. Sistema apresenta ao utilizador os detalhes da espécie: Nome comum, nome científico, imagem, descrição e se tem características predatórias
		5. Sistema seleciona os avistamentos da espécie nas câmaras que o utilizador definiu nas definições como pretendidas
		6. Sistema apresenta ao utilizador os avistamentos da espécie
		7. Sistema gera um gráfico do número de avistamentos por unidade temporal baseada no intervalo utilizado
Fluxo alternativo 1: [Utilizador tinha selecionado um intervalo temporal] (passo 5)		5.1 Sistema filtra os avistamentos que se inserem no intervalo temporal selecionado e ordena-os cronologicamente
		5.2 Regressar ao passo 6

Figura 15.2: Consultar Espécies

Use Case		
Use Case:	Explorar Câmara	
Ator:	Utilizador	
Descrição:	Permite ao utilizador ver em detalhe os avistamentos capturados por uma câmara	
Pré-condição:	Utilizador está autenticado e selecionou a câmara no dashboard principal	
Pós-condição:	Detalhes da câmara são apresentados	
	Ator	Sistema
Cenário Normal	1. Utilizador seleciona câmara para visualizar	
		2. Sistema consulta, na base de dados, a câmara e os avistamentos por ela capturados
		3. Sistema filtra os avistamentos, de forma a apenas apresentar avistamentos que contenham espécies que o utilizador escolheu seguir
		4. Sistema ordena os avistamentos por ordem cronológica
		5. Sistema devolve informação sobre a câmara (nome e localização), assim como até X avistamentos válidos (sendo X um número a determinar após testes de performance da aplicação)
		6. Sistema gera um gráfico do número de avistamentos por unidade temporal baseada no intervalo utilizado
		7. Sistema apresenta ainda um gráfico detalhando o número de avistamentos por unidade de tempo
Fluxo alternativo 1: [Utilizador tinha selecionado um intervalo temporal] (passo 4)		3.1 Sistema filtra os avistamentos que se inserem no intervalo temporal selecionado e ordena-os cronologicamente
		3.2 Regressar ao passo 5

Figura 15.3: Consultar Câmara

Use Case		
Use Case:	Alterar definições	
Ator:	Utilizador Base	
Descrição	Permite ao utilizador alterar definições associadas à sua conta	
Pré-condição:	Utilizador está autenticado	
Pós-condição:	Definições do utilizador são atualizadas	
	Ator	Sistema
Cenário Normal	1. O Utilizador seleciona a opção de alterar definições	
		2.0 Sistema apresenta uma página ao utilizador dividida em 3 secções: - Informações da conta - Câmaras - Espécies
	3. Na secção de informações da conta, o utilizador pode alterar o seu username, se pretende receber o resumo diário, qual o meio onde o receber (email ou telemóvel).	
	4. Na secção das câmaras, o utilizador seleciona quais câmaras pretende observar no dashboard principal e receber no seu resumo diário	
	5. Na secção das espécies, o utilizador seleciona quais as espécies cujos avistamentos lhe interessam	

Figura 15.4: Consultar Câmara

Use Case		
Use Case:	Receber resumo diário	
Ator:	Utilizador Base	
Descrição	Utilizador recebe pelo seu meio definido um resumo dos avistamentos capturados nesse dia	
Pré-condição:	Utilizador ativou a opção de receber o resumo diário	
Pós-condição:	Sistema envia o resumo diário	
	Ator	Sistema
Cenário Normal		1. Sistema lança um CRON job diário para o envio dos resumos diários
		2. Quando o CRON job é ativado, o sistema seleciona todos os utilizadores que assinalaram que pretendiam receber o resumo diário
		3. Sistema seleciona, por cada utilizador, os avistamentos nas câmaras que o utilizador tem nas definições e que contém as espécies que o utilizador pretende seguir
		4. Sistema filtra os avistamentos para que apenas sejam selecionados os avistamentos que ocorreram no dia
		5. Sistema cria um texto de resumo desses avistamentos
		6. Sistema verifica qual o meio de notificação escolhido pelo utilizador (email/telemóvel)
		7. Sistema contacta a API de notificação externa correspondente ao meio de notificação, enviando o texto resumo para este ser enviado ao utilizador

Figura 15.5: Receber Resumo Diário

Use Case		
Use Case:	Autenticar Utilizador	
Ator:	Utilizador	
Descrição	Permite ao utilizador autenticar-se na aplicação	
Pré-condição:	O utilizador não está autenticado	
Pós-condição:	O utilizador está autenticado	
	Ator	Sistema
Cenário Normal	1. Utilizador insere as suas credenciais: email e password	
		2. Sistema valida as credenciais
		3. Sistema redireciona o utilizador para a página principal
Exceção 1 [Credenciais erradas] (Passo 2)		2.1. Sistema informa utilizador que as credenciais não são válidas

Figura 15.6: Autenticar utilizador

Use Case		
Use Case:	Logout	
Ator:	Utilizador	
Descrição	Permite ao utilizador terminar a sua sessão na aplicação	
Pré-condição:	O utilizador está autenticado	
Pós-condição:	A autenticação do utilizador é removida	
	Ator	Sistema
Cenário Normal	1. Utilizador seleciona a opção de logout	
		2. Sessão do utilizador é eliminada no sistema.
		3. Sistema redireciona o utilizador para o ecrã de login

Figura 15.7: Logout

Use Case		
Use Case:	Registar Utilizador	
Ator:	Utilizador Base	
Descrição	Permite a um utilizador registar-se na aplicação	
Pré-condição:	TRUE	
Pós-condição:	Utilizador está registado e autenticado	
	Ator	Sistema
Cenário Normal	1. Utilizador insere as suas credenciais: email, nome, número de telemóvel (opcional), password e confirmação de password	
	2. Utilizador indica se pretende receber o resumo diário e, caso afirmativo, selecionar se pretende receber no email, ou número de telemóvel	
		3. Sistema valida as credenciais: verifica que o email não está já registado e que a password segue os requisitos mínimos
		4. Sistema cria o utilizador com as credenciais fornecidas
		5. Sistema autentica o utilizador na aplicação.
Exceção 1 [Credenciais inválidas] (Passo 3)		6. Sistema redireciona o utilizador para o dashboard principal
		3.1. Sistema informa utilizador que as credenciais não são válidas

Figura 15.8: Logout

Use Case		
Use Case:	Editar câmara	
Ator:	Administrador está autenticado	
Descrição	Permite ao administrador editar informação relativa à câmara selecionada	
Pré-condição:	Administrador está autenticado e na página de visualização da câmara	
Pós-condição:	Informação relativa à câmara é atualizada	
	Ator	Sistema
Cenário Normal	1. Administrador escolhe a opção de editar a câmara	
		2. Sistema apresenta ao utilizador um formulário contendo os vários campos de informação relativos à câmara.
	3. Administrador altera os campos que desejar, submetendo o formulário	
		4. Sistema atualiza o registo da câmara existente na base de dados
Fluxo alternativo 1: [Acesso à câmara passou a ser restrito] (passo 3)		3.1 Sistema remove todas as associações de preferência entre utilizadores base e esta câmara
		3.2 Regressar ao passo 4

Figura 15.9: Logout

Use Case		
Use Case:	Adicionar câmara	
Ator:	Administrador	
Descrição:	Permite ao utilizador registar uma nova câmara no sistema	
Pré-condição:	Administrador está autenticado	
Pós-condição:	Uma nova câmara é adiciona ao sistema	
	Ator	Sistema
Cenário Normal	1. Administrador seleciona operação de adicionar uma nova câmara.	
		2. Sistema apresenta um formulário contendo os vários campos relativos a uma câmara: nome, latitude, longitude, se está numa zona restrita e se a sua visualização está restrita a administradores
	3. Administrador preenche os vários campos submetendo o formulário	
		4. Sistema valida o formulário
		5. Sistema atribui um novo id à câmara criada e insere-a na base de dados
Exceção 1 [Formulário está inválido] (Passo 4)		4.1. Sistema avisa o utilizador dos campos inválidos no formulário.
		4.2. Regressao ao ponto 3

Figura 15.10: Adicionar Câmara

Use Case		
Use Case:	Adicionar espécie	
Ator:	Administrador	
Descrição:	Permite ao utilizador registar uma nova espécie no sistema	
Pré-condição:	Administrador está autenticado	
Pós-condição:	Uma nova espécie é adiciona ao sistema	
	Ator	Sistema
Cenário Normal	1. Administrador seleciona operação de adicionar uma nova espécie.	
		2. Sistema apresenta o formulário de adição de espécie
	3. Administrador preenche os vários campos submetendo o formulário	
		4. Sistema valida o formulário
		5. Sistema atribui um novo id à espécie criada e insere-a na base de dados
Exceção 1 [Formulário está inválido] (Passo 4)		4.1. Sistema avisa o utilizador dos campos inválidos no formulário.
		4.2. Regressao ao ponto 3

Figura 15.11: Adicionar Espécie

Use Case		
Use Case:	Registrar avistamento	
Ator:	Gerador de dados	
Descrição	Permite ao gerador de dados registrar um avistamento	
Pré-condição:	Gerador de dados pretende enviar um registo de avistamento	
Pós-condição:	Avistamento fica registado na aplicação e toda a logística associada é aplicada	
	Ator	Sistema
Cenário Normal	1. Gerador escolhe um id de uma câmara e um id de uma espécie aleatoriamente	
	2. Gerador escolhe aleatoriamente o número correspondente à quantidade de indivíduos avistados	
	3. Gerador envia pedido POST ao endpoint da aplicação principal responsável por registar avistamentos, contendo o id da câmara, o id da espécie e o número de indivíduos no corpo do pedido	
		4. Sistema recebe pedido, validando a existência do id da câmara e do id da espécie.
		5. Sistema cria um novo registo na tabela dos avistamentos, guardando o id da câmara, id da espécie e número de indivíduos avistados, atribuindo um id ao registo e registando o momento de captura.
		6. Sistema verifica que o avistamento não configura nenhuma situação de perigo
		7. <<Include>>: notificar utilizadores de avistamento
Exceção 1 [ID inexistente] (Passo 4)		3.1. Sistema devolve uma resposta json, com o código 404 e uma mensagem de erro a indicar que o id da câmara ou da espécie não existe.
Fluxo alternativo 1: [Ser humano foi detetado em zona restrita Animal predador em zona com humanos] (passo 6)		6.1.1 <<include>> Notificar administradores de situação de perigo
		6.1.2 Regressar ao ponto 7

Figura 15.12: Registrar Avistamento

Use Case		
Use Case:	Inicializar o gerador de dados	
Ator:	Gerador de dados	
Descrição	Permite ao gerador de dados obter informação sobre espécies e câmaras para poder começar a gerar avistamentos	
Pré-condição:	TRUE	
Pós-condição:	Gerador de dados contém a informação pedida	
	Ator	Sistema
Cenário Normal	1. Gerador envia pedido GET ao endpoint da aplicação responsável por devolver informação sobre as espécies	
		2. Sistema consulta a tabela das espécies, gerando uma lista de pares (Id da espécie, nome comum da espécie).
		3. Sistema converte a lista de pares para o formato JSON, devolvendo-a como resposta ao pedido do gerador
	4. Gerador aceita a resposta no formato JSON e guarda a informação das espécies	
	5. Gerador envia pedido GET ao endpoint da aplicação responsável por devolver informação sobre as câmaras existentes no parque	
		6. Sistema consulta a tabela na base de dados que guarda informação relativa às câmaras, gerando uma lista de pares (Id da câmara, nome da câmara).
		7. Sistema converte a lista de pares para o formato JSON, devolvendo-a como resposta ao pedido do gerador
	8. Gerador aceita a resposta no formato JSON e guarda a informação das câmaras	

Figura 15.13: Inicializar o gerador de dados

Use Case		
Use Case:	Notificar utilizadores de avistamento	
Ator:	Gerador de dados	
Descrição:	Notifica os vários utilizadores da aplicação de um novo avistamento	
Pré-condição:	Avistamento foi registado	
Pós-condição:	Utilizadores que pretendiam seguir a espécie e câmara foram notificados	
	Ator	Sistema
Cenário Normal	1. O Gerador de Dados envia notificação de um novo avistamento	
		2. Sistema seleciona, da base de dados, utilizadores que assinalaram que deviam receber notificações da câmara e da espécie do avistamento registado
		3. Sistema envia uma notificação push nova contendo a informação do avistamento

Figura 15.14: Notificar Utilizadores de Avistamento

Use Case		
Use Case:	Notificar administradores de situação de perigo	
Ator:	Gerador de dados	
Descrição:	Administradores são notificados de uma situação de perigo	
Pré-condição:	Situação de perigo ocorreu	
Pós-condição:	Administradores recebem notificação nos seus meios de comunicação pretendidos	
	Ator	Sistema
Cenário Normal	1. O Gerador de Dados envia notificação de um novo avistamento que corresponde a uma das situações de perigo	
		2. Sistema envia notificação push para o browser de todos os administradores indicando a situação de perigo e aconselhando o envio de uma equipa especialista
		3. Por cada administrador, o sistema verifica qual é o seu meio de notificação pretendido: email, telemóvel, ou ambos, recolhendo também o valor do email e/ou o número de telemóvel
		4. Sistema envia pedido aos serviços de notificação externos (serviço de email e/ou de sms), contendo o corpo da mensagem a enviar, assim como o email e/ou número de telemóvel destino.

Figura 15.15: Notificar Administradores de situação de perigo

Use Case		
Use Case:	Gerir Utilizadores	
Ator:	Root user	
Descrição:	Permite ao root user gerir as permissões de cada utilizador	
Pré-condição:	Root está autenticado	
Pós-condição:	Permissões de um utilizador são atualizadas	
	Ator	Sistema
Cenário Normal	1. Seleciona página de gestão de utilizadores	
		2. Sistema apresenta todos os utilizadores
		3. Clicar num utilizador abre uma janela com os detalhes deste utilizador
	4. Root promove um utilizador a administrador	
Fluxo alternativo 1: [Root remove um utilizador de administrador] (passo 4)		5. Sistema atribui a role 'admin' ao utilizador selecionado
		4.1 Sistema remove a role 'admin' do utilizador selecionado

Figura 15.16: Gerir Utilizadores