



Universidade do Minho

UMinho

Mestrado Engenharia Informática
Arquiteturas Aplicacionais
Sistemas Interativos Confiáveis
(2022/23)

PARKFINDER

GRUPO 1

PG50166 Alexandre Daniel Canhoto Poiares Coimbra Soares
PG50168 Alexandre Silva Martins
PG50560 Luís Francisco Ferreira Faria
PG50342 Diogo Viana Ramos Casal Novo

Braga, 22 de junho de 2023

Conteúdo

1 O propósito do Projeto	8
1.1 Contextualização	8
1.2 Definição do Produto	8
1.3 Motivação e Objetivos	9
2 Utilizadores do Produto	10
2.1 Condutor	10
2.2 Administrador de Parques	10
2.3 Gerente	10
2.4 Programador	10
2.5 As prioridades atribuídas aos usuários	11
3 Restrições do Projeto	12
3.1 Restrições Obrigatórias da Solução	12
3.2 Ambiente de Implementação do Sistema	12
3.3 Restrições Prazo/Agendamento	12
3.4 Restrições Orçamento	12
4 Convenções de nomenclatura e definições	13
4.1 Modelo de Domínio	13
4.2 Definição de entidades	13
5 Âmbito do Produto	15
5.1 Diagrama de <i>Use Cases</i>	15
5.2 Atores	18
5.3 Breve Descrição dos Use Cases	19
5.3.1 Login Condutor	19
5.3.2 Registar Condutor	19
5.3.3 Editar Perfil	20
5.3.4 Reserva Agendada	20
5.3.5 Reserva Instantânea	21
5.3.6 Adicionar Lugares a um Parque	21
5.3.7 Remover Lugares de um Parque	22
5.3.8 Ativar Lugares de um Parque	22
5.3.9 Desativar Lugares de um Parque	23
5.3.10 Verificar Reserva	23
5.3.11 Ver Estatísticas	24
5.3.12 Atribuir Administrador a Parque	24
5.3.13 Fechar Parque	25
5.3.14 Criar Parque	25
5.3.15 Criar Gestor	26
5.3.16 Ver Informação de Utilização da Aplicação	26

5.3.17	Atribuir Gestor a Parque	27
5.3.18	Listar Todos os Parques	27
5.3.19	Procurar Parque	27
5.4	Mockups	29
5.4.1	Página de registo	29
5.4.2	Página de login	29
5.4.3	Perfil do Condutor	30
5.4.4	Página Principal	30
5.4.5	Informação do parque	31
6	Requisitos Funcionais	32
6.1	Modelação de requisitos	32
6.2	Requisitos sobre Condutores	33
6.3	Requisitos sobre Programador	38
6.4	Requisitos sobre Gestor	43
6.5	Requisitos sobre Administrador	47
6.6	Requisitos de Sistema	49
7	Requisitos Não Funcionais	51
7.1	Appearance	51
7.2	Usability	52
7.3	Performance	54
7.4	Operational	56
7.5	Maintenance and Support	56
7.6	Security	57
7.7	Legal	58
8	Arquitetura da Solução	59
8.1	Diagramas Complementares da Model Driven Arquitecture	59
8.2	Design Patterns Usados na Solução	63
8.2.1	Facade	63
8.2.2	Template	63
8.2.3	Observer	63
8.3	Arquitetura Final da Aplicação	64
8.4	Instalação/Deployment	65
9	Avaliação de interface e princípios de usabilidade	66
9.1	Condutor	67
9.1.1	Barra de Navegação	67
9.1.2	Página Principal	67
9.1.3	Perfil	68
9.1.4	Página De Parque Detalhada	72
9.2	Administrador	73
9.2.1	Cabeçalho	73
9.2.2	Página Principal	73
9.2.3	Página De Parque Detalhada	74
9.3	Gestor	76
9.3.1	Página Principal	76
9.3.2	Página De Parque Detalhada	78
9.3.3	Página De Criação de Administrador	80
9.4	Programador	80
9.4.1	Página Principal	80

9.5 Comentário sobre as Shneiderman's 8 Golden Rules	82
9.6 Planos para o futuro	82
10 Avaliação da Solução	84

Listas de Figuras

4.1	Modelo de Domínio	13
5.1	Diagrama de <i>Use Cases</i>	15
5.2	Diagrama de <i>Use Cases</i> Gerir Lugares Parque expandido do Administrador	16
5.3	Diagrama de <i>Use Cases</i> Gerir Parque expandido do Gestor	16
5.4	Diagrama de <i>Use Cases</i> Gerir Parques expandido do Programador	17
5.5	Diagrama de <i>Use Cases</i> Gerir Gestores expandido do Programador	17
5.6	Diagrama de Atividades	18
5.7	Mockup da página de registo	29
5.8	Mockup da página de login	29
5.9	Mockup do perfil do condutor	30
5.10	Mockup da página principal	30
5.11	Mockup da informação do parque	31
8.1	PIM com lógica dos parques, reservas e notificações.	60
8.2	PIM com lógica dos utilizadores.	61
8.3	PSM com lógica dos parques e das reservas.	61
8.4	PSM com lógica dos utilizadores e das notificações.	62
8.5	Entity Relationship Diagram	63
8.6	Exemplo do método de login no Swagger	65
9.1	Barra de Navegação	67
9.2	Página Principal	67
9.3	Código QR gerado com o ID do condutor.	68
9.4	Aparência de uma reserva agendada.	68
9.5	Perfil do Condutor	69
9.6	Edição da informação pessoal	69
9.7	Mudança da palavra-passe.	70
9.8	Reservas atuais do condutor	70
9.9	Código QR gerado com o ID da reserva.	71
9.10	Histórico dos pagamentos efetuados.	71
9.11	Histórico das reservas efetuadas.	72
9.12	Página de parque detalhada com descrição.	72
9.13	Página de parque detalhada com caraterísticas.	73
9.14	Cabeçalho da página principal do Administrador	73
9.15	Corpo da página principal do Administrador	74
9.16	Opções de filtragem	74
9.17	Edição de lugares do Parque	75
9.18	Opção de parque instantâneo	75
9.19	Leitor de QRCode para admitir entrada no parque.	76
9.20	Página principal do gestor	76
9.21	Gestão dos administradores	77

9.22 Estatísticas dos parques do Gestor	77
9.23 Adicionar parques ao administrador	78
9.24 Estatísticas do parque na página detalhada do mesmo.	78
9.25 Edição da informação do Parque	79
9.26 Edição da informação do parque	79
9.27 Criação de administrador	80
9.28 Página principal programador	80
9.29 Criação de parque	81
9.30 Informação dos gestores	81
9.31 Criação de gestor	82

Listas de Tabelas

5.1	Especificação de <i>Use Case</i> Login Condutor	19
5.2	Especificação de <i>Use Case</i> Registrar Condutor	20
5.3	Especificação de <i>Use Case</i> Editar Perfil	20
5.4	Especificação de <i>Use Case</i> Reserva Agendada	21
5.5	Especificação de <i>Use Case</i> Reserva Instantânea	21
5.6	Especificação de <i>Use Case</i> Adicionar Lugares a um Parque	22
5.7	Especificação de <i>Use Case</i> Remover Lugares de um Parque	22
5.8	Especificação de <i>Use Case</i> Ativar Lugares de um Parque	23
5.9	Especificação de <i>Use Case</i> Desativar Lugares de um Parque	23
5.10	Especificação de <i>Use Case</i> Verificar Reserva	24
5.11	Especificação de <i>Use Case</i> Ver Estatísticas	24
5.12	Especificação de <i>Use Case</i> Atribuir Administrador a Parque	25
5.13	Especificação de <i>Use Case</i> Fechar Parque	25
5.14	Especificação de <i>Use Case</i> Criar Parque	26
5.15	Especificação de <i>Use Case</i> Criar Gestor	26
5.16	Especificação de <i>Use Case</i> Ver Informação de Utilização da Aplicação	27
5.17	Especificação de <i>Use Case</i> Atribuir Gestor a Parque	27
5.18	Especificação de <i>Use Case</i> Listar Todos os Parques	27
5.19	Especificação de <i>Use Case</i> Procurar Parque	28
6.1	Exemplo especificação de Requisitos	32
6.2	Requisito funcional 1.	33
6.3	Requisito funcional 2.	33
6.4	Requisito funcional 3.	34
6.5	Requisito funcional 4.	34
6.6	Requisito funcional 5.	34
6.7	Requisito funcional 6.	35
6.8	Requisito funcional 7.	35
6.9	Requisito funcional 8.	35
6.10	Requisito funcional 9.	36
6.11	Requisito funcional 10.	36
6.12	Requisito funcional 11.	36
6.13	Requisito funcional 12.	37
6.14	Requisito funcional 13.	37
6.15	Requisito funcional 14.	37
6.16	Requisito funcional 15.	38
6.17	Requisito funcional 16.	38
6.18	Requisito funcional 17.	38
6.19	Requisito funcional 18.	39
6.20	Requisito funcional 19.	39
6.21	Requisito funcional 20.	39

6.22 Requisito funcional 21.	40
6.23 Requisito funcional 22.	40
6.24 Requisito funcional 23.	40
6.25 Requisito funcional 24.	41
6.26 Requisito funcional 25.	41
6.27 Requisito funcional 26.	41
6.28 Requisito funcional 27.	42
6.29 Requisito funcional 28.	42
6.30 Requisito funcional 29.	42
6.31 Requisito funcional 30.	43
6.32 Requisito funcional 31.	43
6.33 Requisito funcional 32.	43
6.34 Requisito funcional 33.	44
6.35 Requisito funcional 34.	44
6.36 Requisito funcional 35.	44
6.37 Requisito funcional 36.	45
6.38 Requisito funcional 37.	45
6.39 Requisito funcional 38.	45
6.40 Requisito funcional 39.	46
6.41 Requisito funcional 40.	46
6.42 Requisito funcional 41.	47
6.43 Requisito funcional 42.	47
6.44 Requisito funcional 43.	47
6.45 Requisito funcional 44.	48
6.46 Requisito funcional 45.	48
6.47 Requisito funcional 46.	48
6.48 Requisito funcional 47.	49
6.49 Requisito funcional 48.	49
6.50 Requisito funcional 49.	49
6.51 Requisito funcional 50.	50
7.1 Requisito não funcional 1.	51
7.2 Requisito não funcional 2.	51
7.3 Requisito não funcional 3.	52
7.4 Requisito não funcional 4.	52
7.5 Requisito não funcional 5.	53
7.6 Requisito não funcional 6.	53
7.7 Requisito não funcional 7.	53
7.8 Requisito não funcional 8.	54
7.9 Requisito não funcional 9.	54
7.10 Requisito não funcional 10.	54
7.11 Requisito não funcional 11.	55
7.12 Requisito não funcional 12.	55
7.13 Requisito não funcional 13.	55
7.14 Requisito não funcional 14.	56
7.15 Requisito não funcional 15.	56
7.16 Requisito não funcional 16.	57
7.17 Requisito não funcional 17.	57
7.18 Requisito não funcional 18.	57
7.19 Requisito não funcional 19.	58

1. O propósito do Projeto

1.1 Contextualização

O aumento gradual do número de veículos a circular nas cidades de Portugal, nos últimos anos, tornou mais difícil o processo de encontrar estacionamento nas cidades. Com o âmbito de facilitar a procura de estacionamento nas cidades por parte dos condutores decidimos desenvolver uma aplicação que indica toda a informação que estes podem necessitar sobre parques de estacionamento nas cidades portuguesas.

Uma aplicação de gestão de parques de estacionamento é um software que permite gerir de forma eficiente e organizada vários parques de estacionamento. Esta aplicação pode ser utilizada por empresas, entidades públicas ou privadas, e permite uma gestão mais eficiente dos parques de estacionamento, garantindo um melhor aproveitamento do espaço disponível, redução de custos, melhorias na segurança, e mais facilidade na gestão dos pagamentos.

A aplicação pode também oferecer recursos de monitorização e controlo da ocupação do espaço, com informações em tempo real sobre a disponibilidade de vagas e tempo de permanência dos veículos.

Além disso, a aplicação de gestão de parques de estacionamento pode incluir ferramentas de análise de dados, que permitem aos gestores do parque de estacionamento visualizar dados e métricas importantes para a sua gestão, como a taxa de ocupação, as receitas geradas, a duração média de estacionamento, entre outros.

Em suma, uma aplicação de gestão de parques de estacionamento pode ser uma ferramenta valiosa para melhorar a eficiência e eficácia na gestão dos parques de estacionamento, proporcionando aos utilizadores maior conveniência, segurança e redução de custos.

1.2 Definição do Produto

Este produto é direcionado para todas as pessoas que necessitem de um lugar para estacionar o seu carro na cidade de Braga.

Este produto deve ser capaz de providenciar registo a novos utilizadores, bem como a autenticação de utilizadores que já tenham conta criada.

De seguida, será disponibilizada uma lista com todos os parques de estacionamento da cidade. Depois do utilizador escolher qual o parque onde vai estacionar o carro, este pode pedir à aplicação qual é o percurso que deve seguir para chegar a esse parque. Por último, o utilizador poderá optar por reservar um lugar do parque com antecedência ou, simplesmente, criar uma reserva instantânea quando chegar ao parque.

1.3 Motivação e Objetivos

O *ParkFinder* surge como resposta ao aumento da dificuldade de encontrar lugares de estacionamento na cidade e pretende incentivar o seu uso para encontrar um lugar de estacionamento de forma rápida e eficaz. Pretende-se com o *ParkFinder* fazer um produto simples e intuitivo que seja capaz de manter e cativar novos utilizadores.

2. Utilizadores do Produto

Efectivamente, nesta secção apresentam-se todos os utilizadores relacionados com este produto listando-se as funções de cada um.

2.1 Condutor

- **Nome/Categoria** - Condutor
- **Função** - O condutor poderá registar-se na aplicação e posteriormente utilizar as suas funcionalidades para conseguir estacionar o carro num parque.
- **Experiência no contexto** - Iniciante
- **Experiência Tecnológica** - Iniciante

2.2 Administrador de Parques

- **Nome/Categoria** - Administrador de Parques
- **Função** - O administrador de parques é responsável por gerir os lugares disponíveis dos parques a que está associado.
- **Experiência no contexto** - Mestre
- **Experiência Tecnológica** - Iniciante

2.3 Gerente

- **Nome/Categoria** - Gerente
- **Função** - O gerente vai receber dados estatísticos relacionados com os parques para garantir que o sistema está a dar lucro.
- **Experiência no contexto** - Mestre
- **Experiência Tecnológica** - Iniciante

2.4 Programador

- **Nome/Categoria** - Programador
- **Função** - O programador vai estar encarregue de assegurar o funcionamento da aplicação.
- **Experiência no contexto** - Mestre
- **Experiência Tecnológica** - Mestre

2.5 As prioridades atribuídas aos usuários

- **Utilizadores Principais** - Condutor
- **Utilizadores Secundários** - Administrador de Parques, Gerente e Programador

O sucesso da aplicação depende dos condutores, e, por esse motivo, são nestes que a maioria dos requisitos são centrados. O administrador de parques, o gerente e o programador tem um parecer nos requisitos, no entanto, caso estes colidam com os dos condutores, os condutores vão ter precedência.

3. Restrições do Projeto

3.1 Restrições Obrigatórias da Solução

- **Descrição:** O condutor tem acesso a uma listagem de todos os parques de estacionamento possíveis, e após a seleção do parque será capaz de selecionar a forma de reservar o lugar de estacionamento.

Justificação: De forma a garantir a função principal da aplicação que é possibilitar o estacionamento aos condutores.

- **Descrição:** A aplicação tem de ser um *web site*.

Justificação: A aplicação deverá poder ser usada em qualquer lugar, a partir do seu computador ou *smartphone*.

3.2 Ambiente de Implementação do Sistema

A Empresa irá alocar os recursos necessários para manter a aplicação.

As sessões de teste da aplicação serão realizadas nos *browsers* Google Chrome, Microsoft Edge e Brave.

3.3 Restrições Prazo/Agendamento

- **Descrição:** O documento presente terá de ser entregue juntamente com a aplicação, até ao dia 22 de junho de 2023.

Justificação: De forma a poder ser avaliado o estado final do projeto pela equipa docente.

3.4 Restrições Orçamento

- **Descrição:** O orçamento total para o desenvolvimento do projeto é de 20 000€ (vinte mil euros), durante um período de 3 meses.

Justificação: A equipa responsável pelo desenvolvimento do projeto é constituída por quatro engenheiros de *software*. Para além de ter em conta os salários dos elementos, é preciso também a compra de um domínio e de servidores que sejam capazes de assegurar o funcionamento da aplicação, bem como o armazenamento dos dados da mesma.

4. Convenções de nomenclatura e definições

4.1 Modelo de Domínio

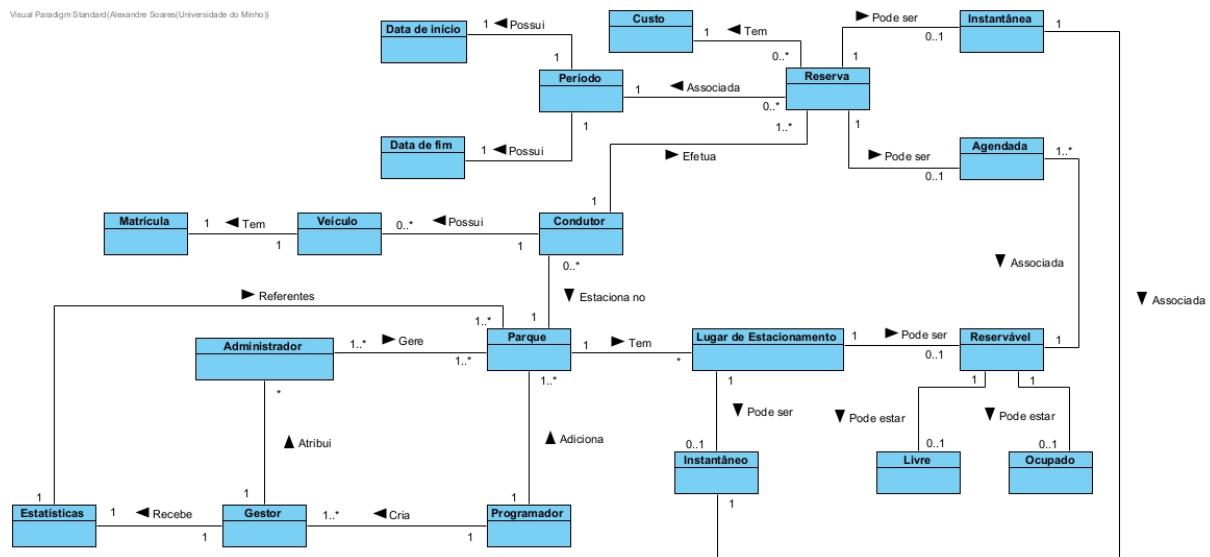


Figura 4.1: Modelo de Domínio

4.2 Definição de entidades

Nesta secção, são abordadas as diferentes entidades presentes no Modelo de Domínio apresentado na secção anterior (figura 5.1).

- **Condutor:** Principal utilizador da aplicação que vai beneficiar diretamente do uso da aplicação;
- **Administrador:** Utilizador encarregue de monitorizar os parques de que está encarregue;
- **Gerente:** Utilizador responsável pela criação de contas de administrador e pela associação dos administradores aos parques. Recebe também dados estatísticos sobre o funcionamento dos parques que gere;
- **Programador:** Utilizador encarregue de garantir o 'bom' funcionamento da aplicação. Vai ser capaz de criar contas de Gestor e vai poder também adicionar novos parques à aplicação;
- **Parque:** Representa um parque de estacionamento e toda a informação relevante sobre o mesmo, como por exemplo, os seus lugares disponíveis;
- **Lugar de Estacionamento:** O lugar de estacionamento representa um dos lugares individuais de estacionamento de um determinado parque;

- **Reservável:** A entidade reservável serve para indicar que um lugar de estacionamento deste tipo só aceita reservas agendadas.
- **Instantâneo:** A entidade instantâneo serve para indicar que um lugar de estacionamento deste tipo só aceita reservas instantâneas.
- **Livre:** A entidade livre indica que um lugar de estacionamento reservável está livre;
- **Ocupado:** A entidade ocupado indica que um lugar de estacionamento reservável está ocupado;
- **Estatísticas:** As estatísticas são informações fornecidas aos gestores sobre alguns dados relevantes do funcionamento dos parques que gerem, como por exemplo, montantes totais pagos, volumes de estacionamentos, entre outros;
- **Veículo:** O veículo é possuído por um condutor e vai ocupar um lugar de estacionamento quando estiver no parque;
- **Matrícula:** A matrícula representa o código identificador único de um veículo;
- **Reserva:** A reserva é o que vincula uma conta de utilizador a um lugar do parque de estacionamento;
- **Instantânea:** Tipo de reserva que é efetuada pouco antes de estacionar o veículo no parque;
- **Agendada:** Tipo de reserva que é efetuada com alguma antecedência antes de estacionar o veículo no parque;
- **Custo:** Custo associado à ocupação de um lugar do parque. Vai depender do tempo da reserva e do tipo da reserva;
- **Período:** A entidade período é utilizada para guardar o tempo associado o determinada reserva;
- **Data de Início:** A data de inicio corresponde ao dia, à hora e aos minutos de início de uma reserva;
- **Data de Fim:** A data de fim corresponde ao dia, à hora e aos minutos de fim de uma reserva;

5. Âmbito do Produto

5.1 Diagrama de *Use Cases*

De maneira a compreender melhor o contexto do sistema, vai ser apresentado um diagrama de *Use Cases*. Neste vão ser explicitadas algumas das principais funcionalidades do sistema, bem como os atores do mesmo. Neste diagrama, é ainda possível identificar, quais as funcionalidades a que cada tipo de ator do sistema vai ter acesso.

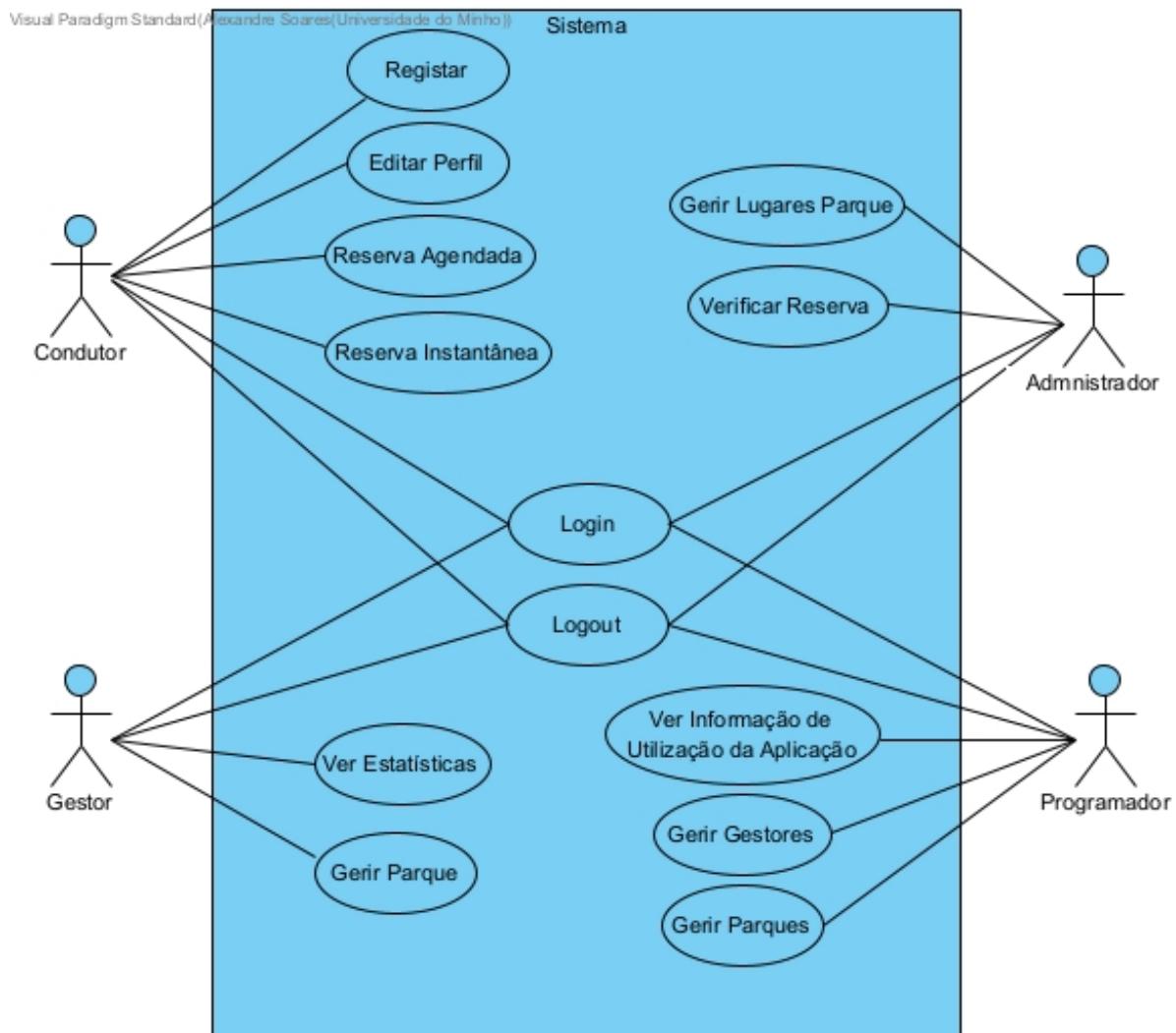


Figura 5.1: Diagrama de *Use Cases*

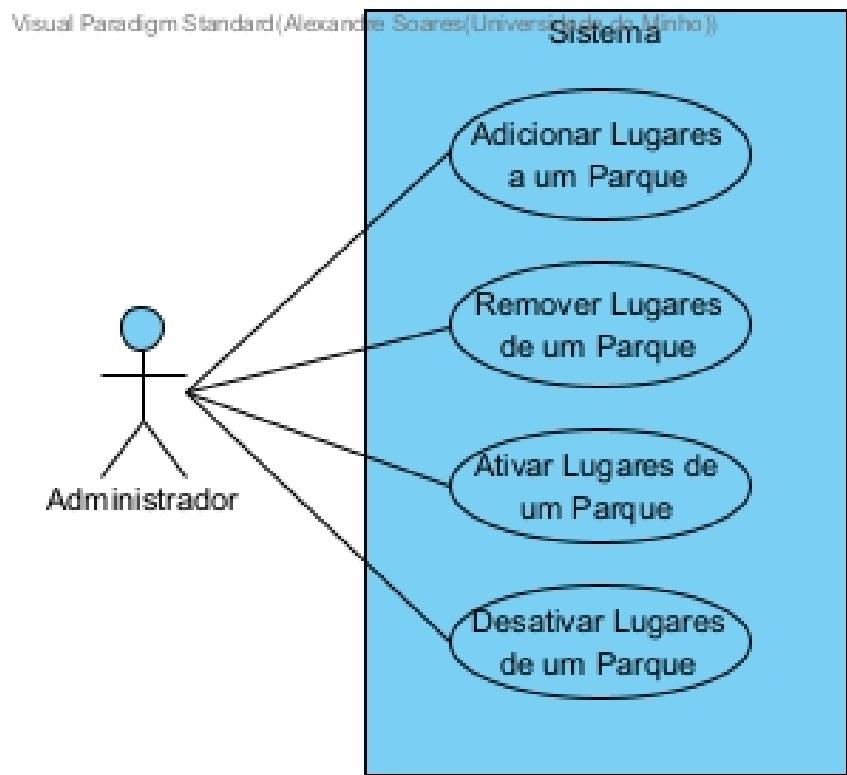


Figura 5.2: Diagrama de *Use Cases* Gerir Lugares Parque expandido do Administrador

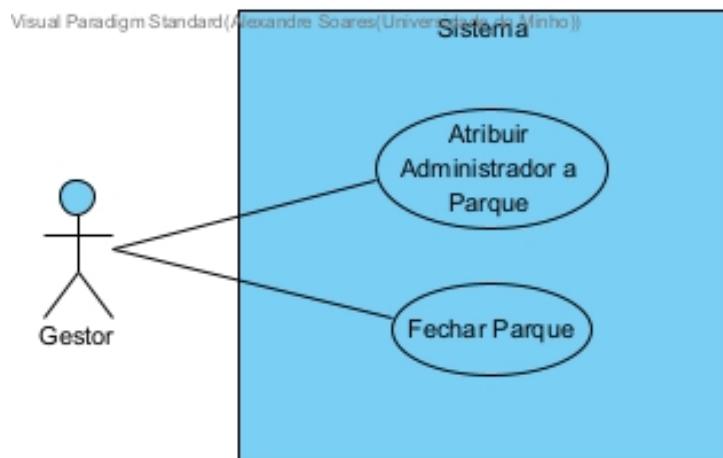


Figura 5.3: Diagrama de *Use Cases* Gerir Parque expandido do Gestor

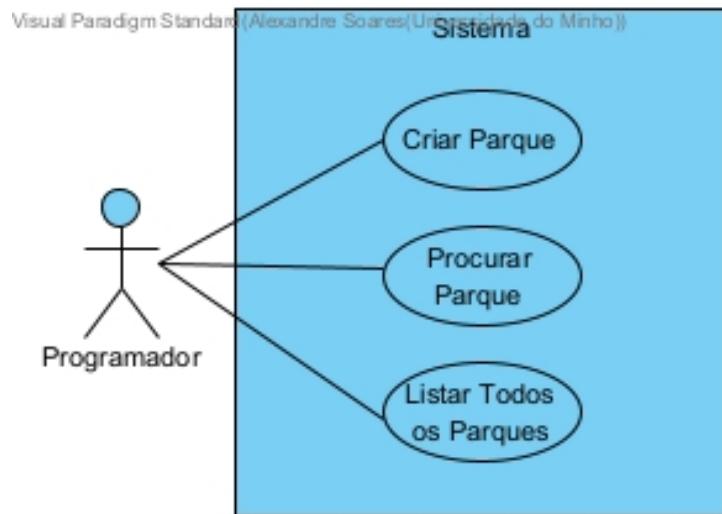


Figura 5.4: Diagrama de *Use Cases* Gerir Parques expandido do Programador

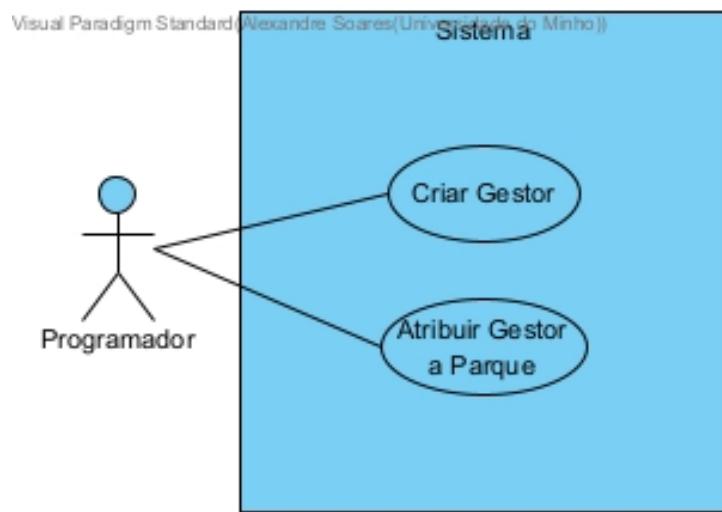


Figura 5.5: Diagrama de *Use Cases* Gerir Gestores expandido do Programador

5.2 Atores

Como está representado no diagrama anterior, o nosso sistema suporta quatro tipos de atores: o Condutor, o Administrador, o Gerente e o Programador.

O **Condutor** é o utilizador comum do nosso sistema, vai ter acesso a todas as funcionalidades que a nossa aplicação vai disponibilizar para o público geral.

O **Administrador** é o indivíduo responsável por garantir o correto funcionamento dos parques que administra. Para tal ser possível vai ter acesso a informação relevante dos parques, vai ser capaz de ativar/desativar lugares aos parques e, para além disso, vai ser capaz de fazer login e logout da aplicação.

O **Gerente** é o indivíduo que através das informações estatísticas que recebe dos parques que gera vai controlar a alocação dos administradores aos parques. Para além disso, vai ser capaz de criar contas de administrador, fazer login e logout da aplicação.

O **Programador** é o indivíduo responsável por garantir que a aplicação mantém o seu bom funcionamento. Para tal, este vai ter acesso às estatísticas globais da aplicação, vai puder criar/eliminar contas de gestores, vai associar/desassociar os gestores aos parques, vai conseguir adicionar/remover parques de estacionamento da aplicação e vai puder aceder à lista de todos os parques e gestores da aplicação. Para além disso, vai ser capaz de fazer login e logout da aplicação.

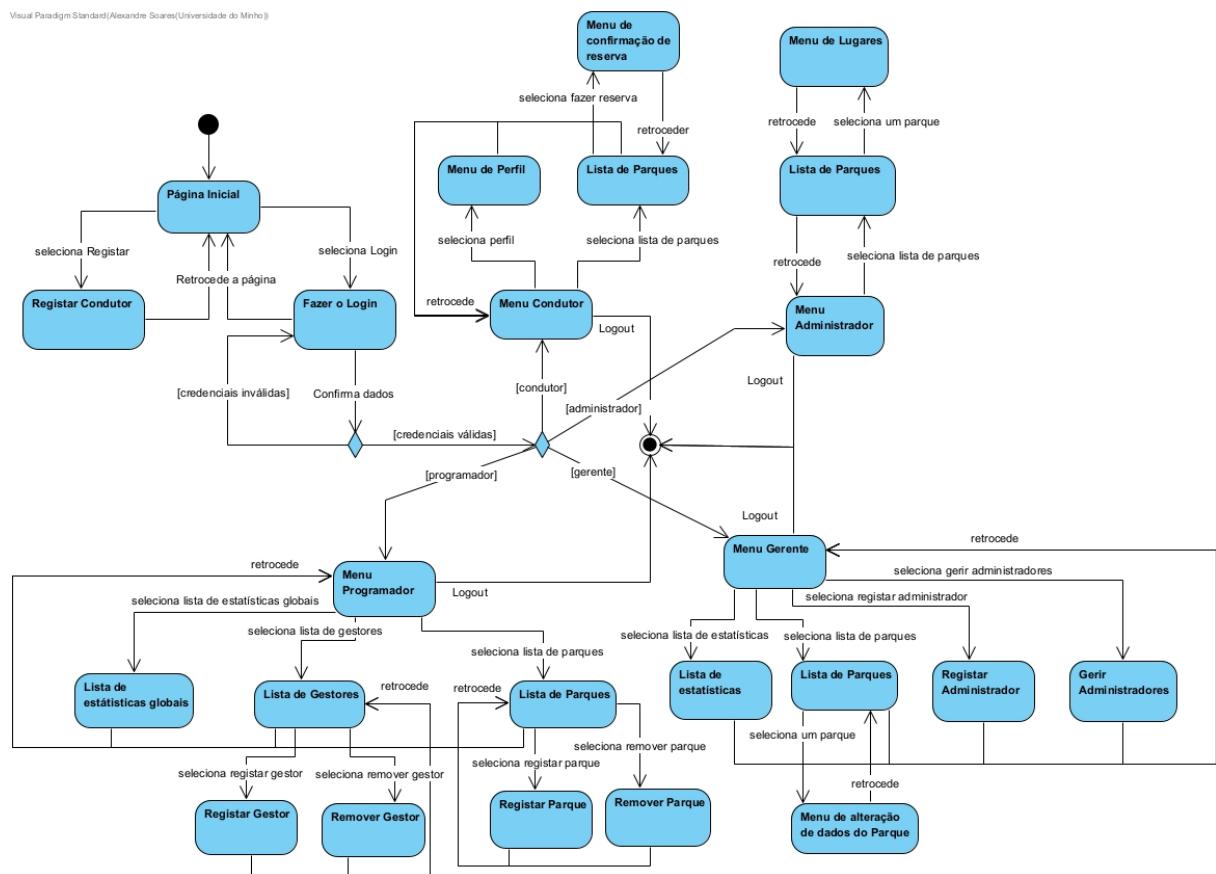


Figura 5.6: Diagrama de Atividades

5.3 Breve Descrição dos Use Cases

Nesta secção será apresentada uma especificação tabelar de cada *Use Case* considerado, de modo a facilitar todo o processo de implementação de cada funcionalidade do nosso sistema.

Deste modo, consideramos que é bastante perceptível o fluxo sequencial da interacção do ator com o sistema.

5.3.1 Login Condutor

O *use case* Login Condutor consiste na autenticação de um condutor que já efetuou o registo e, portanto, já se encontra na base de dados do sistema. Para o *login* ser efetuado vai necessitar dos seguintes parâmetros nome de utilizador e palavra-passe.

Os casos de exceção deste *use case* acontecem quando o nome do utilizador não se encontra no sistema e quando a palavra-passe introduzida não corresponde aquela guardada no sistema.

USE-CASE:	Login Condutor	
Ator:	Condutor	
Pré-Condição:	Condutor está registado no sistema	
Pós-Condição:	Condutor está autenticado no sistema	
Cenário Normal	Ator Input	System Response
	2 - Fornece credenciais.	1 - Pede credenciais necessárias para o login. 3 - Valida credenciais. 4 - Dirige o Condutor para a página principal.
Exceção 1 : [username inexistente] (Passo 3)		3.1 - Informa o Condutor que não existe tal username.
Exceção 2 : [password inválida] (passo 3)		3.2 - Informa o Condutor que a password não corresponde ao username.

Tabela 5.1: Especificação de *Use Case* Login Condutor

5.3.2 Registar Condutor

O *use case* Registar Condutor consiste no ato de registar um condutor que ainda não está presente no sistema. Para o registo ser efetuado vão precisar de ser preenchidos os campos: nome próprio completo, NIF(número de identificação fiscal), género, número de telemóvel, endereço de email, palavra-passe e confirmação da mesma. Os casos de erro acontecem quando o sistema deteta que o email inserido já existe no sistema ou quando a palavra-passe introduzida não cumpre os requisitos de uma palavra-passe válida.

USE-CASE:	Registrar Condutor	
Autor:	Condutor	
Pré-Condição:	Condutor não está registado no sistema	
Pós-Condição:	Condutor está registado no sistema	
Cenário Normal	Autor Input	System Response
	2 - Fornece informações.	1 - Pede informações necessárias para o registo. 3 - Valida informações. 4 - Informa que o Condutor está registado.
Exceção 1 : [email existente] (Passo 3)		3.1 - Informa o Condutor que o email já existe no sistema.
Exceção 2 : [password inválida] (passo 3)		3.2 - Informa o Condutor que a password não cumpre os requisitos necessários.

Tabela 5.2: Especificação de *Use Case* Registrar Condutor

5.3.3 Editar Perfil

O *use case* Editar Perfil consiste na alteração dos campos editáveis do perfil. Para tal ser possível o sistema vai apresentar a informação do perfil do Condutor em formato de edição para que o mesmo a possa alterar.

USE-CASE:	Editar Perfil	
Autor:	Condutor	
Pré-Condição:	Condutor está autenticado no sistema	
Pós-Condição:	O perfil está editado	
Cenário Normal	Autor Input	System Response
	1 - Seleccionar opção de editar perfil 3 - Fornece a nova informação. 5 - Confirma a mudança	2 - Mostra as informações atuais do Condutor. 4 - Valida as informações 6 - Muda as informações. 7 - Mostra as informações atuais do Condutor.
Exceção 1 : [entrada inválida] (Passo 3)		3.1 - Informa o Condutor que a entrada não é válida e as suas razões.

Tabela 5.3: Especificação de *Use Case* Editar Perfil

5.3.4 Reserva Agendada

O *use case* Reserva Agendada permite ao condutor reservar um lugar de estacionamento livre num parque à sua escolha. Este tipo de reserva tem de ser efetuado com alguma antecedência, mas têm o benefício do seu ser mais reduzido quando comparado com o outro tipo de reserva.

USE-CASE:	Reserva Agendada	
Ator:	Condutor	
Pré-Condição:	Condutor está autenticado no sistema	
Pós-Condição:	Reserva Agendada guardada no sistema	
Cenário Normal	Autor Input 1 - Seleciona a opção de fazer uma reserva 3 - Fornece informações. 6- Conclui o pagamento.	System Response 2 - Pede informações necessárias para a reserva. 4 - Valida informações. 5 - Notifica sobre o pagamento. 7 - Verifica o pagamento. 8- Informa que a reserva está concluída.
Exceção 1 : [Lugares preenchidos] (Passo 1)	0.1 - Volta para a lista de parques 0.2 - Escolhe um parque diferente.	

Tabela 5.4: Especificação de *Use Case* Reserva Agendada

5.3.5 Reserva Instantânea

O *use case* Reserva Instantânea permite ao condutor reservar um lugar de estacionamento livre num parque à sua escolha. Este tipo de reserva pode ser efetuado na hora de chegada ao parque, mas o custo da reserva é superior ao da reserva agendada.

USE-CASE:	Reserva rápida	
Ator:	Condutor	
Pré-Condição:	Condutor está autenticado no sistema	
Pós-Condição:	Reserva Instantânea guardada no sistema	
Cenário Normal	Autor Input 1 - Selecionar opção de reserva instantânea de um parque. 4 - Confirma a reserva.	System Response 2 - Faz uma reserva instantânea de um lugar vazio nesse parque. 3 - Informa que Condutor sobre a reserva.
Exceção 1 : [Lugares preenchidos] (Passo 1)		1.1 - Informa o Condutor que não é possível fazer a reserva.
Exceção 2 : [Condutor cancela (passo 3)]		3.1 - Volta á página principal.

Tabela 5.5: Especificação de *Use Case* Reserva Instantânea

5.3.6 Adicionar Lugares a um Parque

O *use case* Adicionar Lugares a um Parque consiste em aumentar o número de lugares disponíveis num parque. Para isso, existe a opção de adicionar lugares na página que representa um parque.

USE-CASE:	Adicionar Lugares a um Parque	
Autor:	Administrador	
Pré-Condição:	Administrador está autenticado no sistema	
Pós-Condição:	Lugares foram adicionados ao parque no sistema	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de listar parques. 3 - Seleciona o parque pretendido. 5 - Seleciona a opção de adicionar lugares. 7 - Fornece o número de lugares a adicionar ao parque.	2 - Mostra a lista de parques. 4 - Mostra a página do parque. 6 - Pede o número de lugares a acrescentar. 8 - Altera o registo dos lugares do parque.

Tabela 5.6: Especificação de *Use Case* Adicionar Lugares a um Parque

5.3.7 Remover Lugares de um Parque

O *use case* Remover Lugares de um Parque consiste em diminuir o número de lugares disponíveis num parque. Para isso, existe a opção de remover lugares na página que representa um parque.

USE-CASE:	Remover Lugares de um Parque	
Autor:	Administrador	
Pré-Condição:	Administrador está autenticado no sistema	
Pós-Condição:	Lugares foram removidos do parque no sistema	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de listar parques. 3 - Seleciona o parque pretendido. 5 - Seleciona a opção de remover lugares. 7 - Fornece o número de lugares a remover do parque.	2 - Mostra a lista de parques. 4 - Mostra a página do parque. 6 - Pede o identificador dos lugares a remover. 8 - Verifica que não existem reservas nos lugares especificados. 9 . Altera o registo dos lugares do parque.
Exceção 1 : [Lugar com reservas] (Passo 8)		8.1 - Informa o Administrador que não pode remover esse lugar.

Tabela 5.7: Especificação de *Use Case* Remover Lugares de um Parque

5.3.8 Ativar Lugares de um Parque

O *use case* Ativar Lugares de um Parque consiste, em manualmente, tornar um ou mais lugares do parque disponíveis. Para isso existe a opção de ativar lugares na página de um parque.

USE-CASE:	Ativar Lugares de um Parque	
Ator:	Administrador	
Pré-Condição:	Administrador está autenticado no sistema	
Pós-Condição:	Os lugares selecionados ficaram ativados.	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de listar parques. 3 - Seleciona o parque pretendido. 5 - Seleciona a opção de desativar lugares do parque. 7 - Seleciona os lugares a ativar.	2 - Mostra a lista de parques. 4 - Mostra a página do parque. 6 - Mostra lista de lugares do parque. 8 - Confirma que os lugares selecionados foram ativados.

Tabela 5.8: Especificação de *Use Case* Ativar Lugares de um Parque

5.3.9 Desativar Lugares de um Parque

O *use case* Desativar Lugares de um Parque consiste, em manualmente, tornar um ou mais lugares do parque indisponíveis. Para isso existe a opção de desativar lugares na página de um parque.

USE-CASE:	Desativar Lugares de um Parque	
Ator:	Administrador	
Pré-Condição:	Administrador está autenticado no sistema	
Pós-Condição:	Os lugares selecionados ficaram desativados	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de listar parques. 3 - Seleciona o parque pretendido. 5 - Seleciona a opção de desativar lugares do parque. 7 - Seleciona os lugares a desativar.	2 - Mostra a lista de parques. 4 - Mostra a página do parque. 6 - Mostra lista de lugares do parque. 8 - Confirma que os lugares selecionados foram desativados.

Tabela 5.9: Especificação de *Use Case* Desativar Lugares de um Parque

5.3.10 Verificar Reserva

O *use case* Verificar Reserva consiste em permitir ao administrador verificar o estado de uma reserva que pode ser: pendente pagamento, agendada, ocupada ou concluída. Para verificar uma reserva o administrador vai ter de fornecer ao sistema o código identificador da mesma.

USE-CASE:	Verificar Reserva	
Ator:	Administrador	
Pré-Condição:	Administrador está autenticado no sistema	
Pós-Condição:	Reserva foi verificada	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de listar parques. 3 - Seleciona o parque pretendido. 5 - Seleciona a opção de listar reservas do parque. 7 - Introduz o identificador da reserva a verificar.	2 - Mostra a lista de parques. 4 - Mostra a página do parque. 6 - Mostra lista de reservas do parque. 8 - Mostra informação da reserva.
Exceção 1 : [reserva inexistente] (Passo 8)		8.1 - Informa o administrador que não existe nenhuma reserva com esse identificador.

Tabela 5.10: Especificação de *Use Case* Verificar Reserva

5.3.11 Ver Estatísticas

O *use case* Ver Estatísticas consiste em apresentar ao gestor os dados sobre o uso dos parques de estacionamento que está encarregue. Para isso, existe uma opção para ver essas estatísticas na sua página principal.

USE-CASE:	Ver Estatísticas	
Ator:	Gestor	
Pré-Condição:	Gestor está autenticado no sistema	
Pós-Condição:	É apresentada ao Gestor a tabela de estatísticas	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de observação das estatísticas.	2 - Mostra uma página de informação e estatísticas.

Tabela 5.11: Especificação de *Use Case* Ver Estatísticas

5.3.12 Atribuir Administrador a Parque

O *use case* Atribuir Administrador a Parque consiste em atribuir o controlo de um parque a um administrador. Para isso, o Gestor tem uma opção para o fazer na página do parque em questão.

USE-CASE:	Atribuir Administrador a Parque	
Ator:	Gestor	
Pré-Condição:	Gestor está autenticado no sistema	
Pós-Condição:	Administrador selecionado ficou atribuído ao parque escolhido.	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de listar parques. 3 - Seleciona o parque a que quer atribuir um administrador. 5 - Seleciona a opção de atribuir administrador. 7 - Seleciona o administrador pretendido.	2 - Mostra a lista de parques. 4 - Mostra a página do parque. 6 - Mostra lista de administradores.

Tabela 5.12: Especificação de *Use Case* Atribuir Administrador a Parque

5.3.13 Fechar Parque

O *use case* Fechar Parque consiste em tornar um parque indisponível para reservas e cancelar as reservas futuras. Para isso, irá existir essa opção na página de um parque do Gestor.

USE-CASE:	Fechar Parque	
Ator:	Gestor	
Pré-Condição:	Gestor está autenticado no sistema	
Pós-Condição:	O parque está fechado no sistema	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de listar parques. 3 - Seleciona parque que é para fechar. 5 - Seleciona a opção de fechar parque. 7 - O gestor confirma.	2 - Mostra a lista de parques. 4 - Mostra a página do parque. 6 - Pede ao Gestor para confirmar. 8 - Cancela as reservas futuras. 9 - Notifica os Condutores cujas reservas foram canceladas.

Tabela 5.13: Especificação de *Use Case* Fechar Parque

5.3.14 Criar Parque

O *use case* Criar Parque permite ao Programador adicionar um novo parque ao sistema. Para tal este vai precisar de fornecer a informação necessária para possibilitar a criação do mesmo.

USE-CASE:	Criar Parque	
Autor:	Programador	
Pré-Condição:	Programador está autenticado no sistema	
Pós-Condição:	O novo parque está registado no sistema	
Cenário Normal	Autor Input 1 - Selecionar opção de criar parque. 3 - Fornece as informações necessárias para a criação de um parque.	System Response 2 - Pede informações necessárias para a criação do parque. 4 - Valida informações. 5 - Informa que o parque foi criado com sucesso.
Exceção 1: [localização em uso] (Passo 4)		4.1 - Informa o Programador que um parque nessa localização já existe no sistema.

Tabela 5.14: Especificação de *Use Case* Criar Parque

5.3.15 Criar Gestor

O *use case* Criar Gestor permite ao Programador adicionar um novo gestor ao sistema. Para tal este vai precisar de fornecer a informação necessária para possibilitar a criação do mesmo.

USE-CASE:	Criar Gestor	
Autor:	Programador	
Pré-Condição:	Programador está autenticado no sistema	
Pós-Condição:	O novo Gestor está registado no sistema	
Cenário Normal	Autor Input 1 - Selecionar opção de criar gestor. 3 - Fornece as informações necessárias para a criação de um gestor.	System Response 2 - Pede informações necessárias para a criação do gestor. 4 - Valida informações. 5 - Informa que o gestor foi criado com sucesso.
Exceção 1 : [email existente] (Passo 4)		4.1 - Informa o Programador que um utilizador com esse email já existe no sistema.

Tabela 5.15: Especificação de *Use Case* Criar Gestor

5.3.16 Ver Informação de Utilização da Aplicação

O *use case* Ver Informação de Utilização da Aplicação permite ao Programador visualizar toda a informação relevante do uso da aplicação para que o mesmo possa garantir o 'bom' funcionamento da aplicação.

USE-CASE:	Ver Informação de Utilização da Aplicação	
Ator:	Programador	
Pré-Condição:	Programador está autenticado no sistema	
Pós-Condição:	Programador acedeu e à informação	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de observação da aplicação.	2 - Mostra uma página de informação e estatísticas sobre a utilização da aplicação

Tabela 5.16: Especificação de *Use Case* Ver Informação de Utilização da Aplicação

5.3.17 Atribuir Gestor a Parque

O *use case* Atribuir Gestor a Parque consiste em dar a permissão a um Gestor para atuar sobre um determinado parque.

USE-CASE:	Atribuir Gestor a Parque	
Ator:	Programador	
Pré-Condição:	O gestor está registado no sistema	
Pós-Condição:	Reserva está efetuada	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de listar parques. 3 - Seleciona parque a que quer atribuir um gestor. 5 - Seleciona a opção de atribuir gestor. 7 - Seleciona o gestor pretendido.	2 - Mostra a lista de parques. 4 - Mostra a página do parque. 6 - Mostra lista de gestores.

Tabela 5.17: Especificação de *Use Case* Atribuir Gestor a Parque

5.3.18 Listar Todos os Parques

O *use case* Listar Todos os Parques, consiste em apresentar uma lista de todos os parques, ativos ou não.

USE-CASE:	Listar Todos os Parques	
Ator:	Programador	
Pré-Condição:	Programador está autenticado no sistema.	
Pós-Condição:	Programador acedeu à lista de todos os parques registados.	
Cenário Normal	Autor Input	System Response
	1 - Seleciona a opção de listar parques.	2 - Mostra a lista de parques registados.

Tabela 5.18: Especificação de *Use Case* Listar Todos os Parques

5.3.19 Procurar Parque

O *use case* Procurar Parque consiste em filtrar a lista de parques pela presença do conjunto de caracteres escritos na barra de procura. Para que tal seja possível, a página da lista de parques terá uma barra de procura.

USE-CASE:	Procurar Parque	
Ator:	Programador	
Pré-Condição:	Programador está autenticado no sistema	
Pós-Condição:	Programador encontrou o parque	
Cenário Normal	Ator Input 1 - Seleciona a opção de listar parques. 3 - Escreve o nome do parque. 5 - Seleciona o parque pretendido.	System Response 2 - Mostra a lista de parques e a barra de procura. 4 - Mostra os resultados da procura.
Exceção 1 : [sem resultados] (Passo 4)	5.1 - Fornece um novo nome	4.1 - Não encontrou nenhum resultado para o nome introduzido.

Tabela 5.19: Especificação de *Use Case* Procurar Parque

5.4 Mockups

Após os primeiros contactos com o cliente e terem sido definidas as ideias principais para o sistema, foram criadas algumas Mockups que servem de representação inicial daquilo que será o sistema final.

5.4.1 Página de registo



Figura 5.7: Mockup da página de registo

5.4.2 Página de login

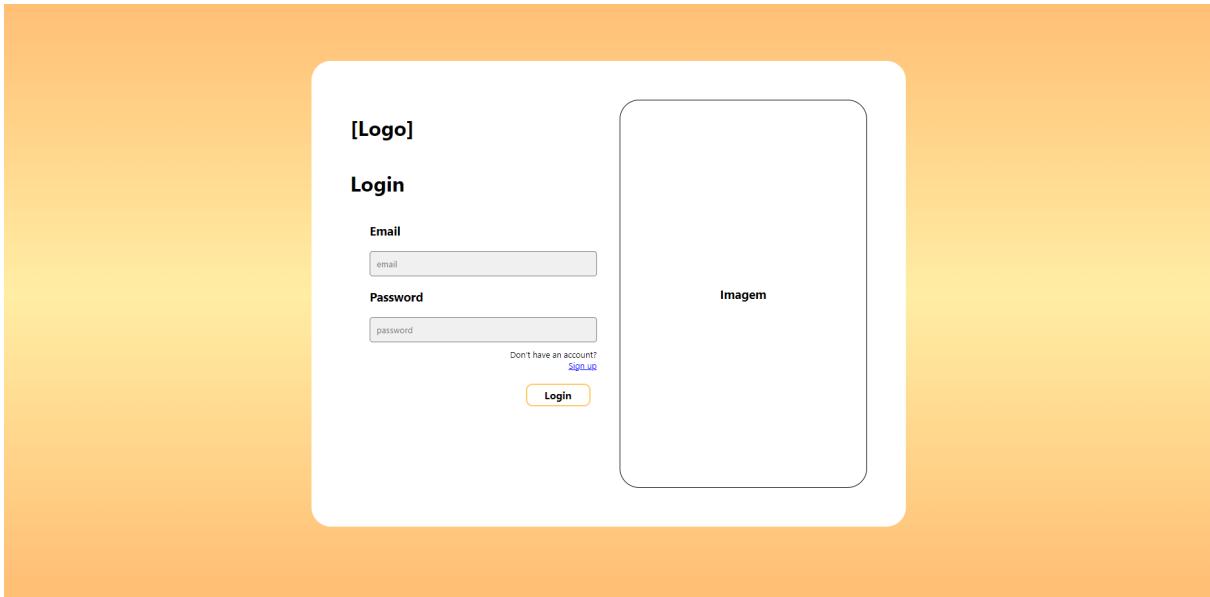


Figura 5.8: Mockup da página de login

5.4.3 Perfil do Condutor

[Logo][appName] [Perfil]

Airports Cities Train Stations [search] [notifyBell] Login

Basic Information

Name: Driver
Email: driverEmail@gmail.com
Phone Number: 987654321

Other Information

Occupation: driverOnFreeTimes
Nacionality: DriverCountry
City: DriverCity
Address: DriverHouse
Postal Code: 1234-567

Password

Old Password:
New Password:
Confirm New Password:

Save

Figura 5.9: Mockup do perfil do condutor

5.4.4 Página Principal

[Logo][appName] [Perfil]

Airports Cities Train Stations [search] [notifyBell] Login

Local De Pesquisa

[SelectDataInicio] [SelectDataFim] Sort by ▾

PARQUE VISCONDE DO RAIO (797 m) 45/96

Estimated cost: €1.30

Imagem

- Public covered parking
- 7 min. walk from the heart of the city
- Accessible from Monday to Friday from 8:00 am to 8:00 pm and Saturdays from 10:00 am to 8:00 pm.

See details Book

B&B BRAGA LAMAÇÃES (2.7 km) 22/51

Estimated cost: €7.00

Imagem

- Covered Hotel Parking
- 10 min. from University of Minho
- taxis service Accessible 24/7

See details Book

BRAGA PARQUE (1.1km) 186/268

Filter Remove all filters

Type of parking

Covered
 Outside

Services

Electric recharge service
 Multiple entries and exits
 Quick access
 Elevator

Reservations

Schedule reservation
 On arrival

Apply filters

Figura 5.10: Mockup da página principal

5.4.5 Informação do parque

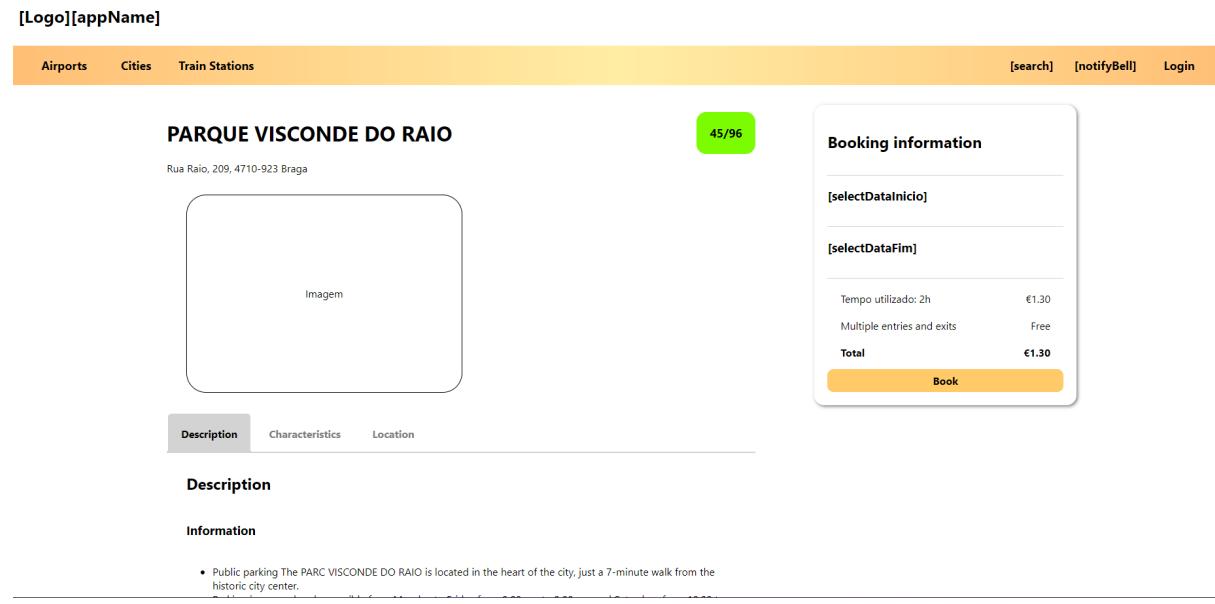


Figura 5.11: Mockup da informação do parque

6. Requisitos Funcionais

6.1 Modelação de requisitos

Para o levantamento de requisitos, foi utilizado como forma de representação, a *requirement shell* do modelo de *Volere*, de forma a descrever concisamente os requisitos.

Requirement:	Type:	Event/Use Case:
Description:		
Rationale:		
Originator:		
Fit Criterion:		
Customer Satisfaction:	Customer Dissatisfaction:	Priority:
Conflicts:		
History:		

Tabela 6.1: Exemplo especificação de Requisitos

Como caracterização da tabela de representação de requisitos, é necessário descrever os campos:

- **Requirement:** Número de identificação do requisito.
- **Requirement Type:** Tipo de requisito, considerando o modelo de *Volere* (Requisito Funcional 9).
- **Event/Use Case:** Número do evento ou Use Case associado.
- **Description:** Descrição, clara e concisa do requisito.
- **Rationale:** Justificação, razão da existência do requisito.
- **Originator:** Quem originou o requisito.
- **Fit Criterion:** Critério em que se insere.
- **Customer Satisfaction:** Apreciação de 1 a 5 em que 1 significa que existe um interesse pequeno que o requisito seja satisfeito, e 5 significa que irão ficar extremamente satisfeitos com a implementação do requisito.
- **Customer Dissatisfaction:** Apreciação de 1 a 5 em que 1 significa que é quase irrelevante que o requisito seja satisfeito, e 5 significa que irão ficar extremamente insatisfeitos com a ausência da correcta implementação do requisito.
- **Priority:** Define o índice de prioridade de implementação de requisitos:
 - **Must:** Requisitos obrigatórios.
 - **Should:** Requisitos que devem ser implementados.

- **Could:** Requisitos que não são necessários, mas são desejados.
- **Won't:** Requisitos que podem ser considerados posteriormente.
- **Conflicts:** Problema encontrado na conceptualização do requisito.
- **History:** Data de criação do requisito.

6.2 Requisitos sobre Condutores

Requirement #1	Type: 9	Event/Use Case:1
Description: Como Condutor, quero poder iniciar sessão com o meu nome de utilizador e palavra-passe.		
Rationale: Para que o Condutor tenha acesso às funcionalidades exclusivas de Condutor que somente ele pode realizar, contribuindo para a segurança da aplicação.		
Originator: Introspeção		
Fit Criterion: Na página de início de sessão, após introduzir as credenciais (email e palavra-passe) de uma conta de Condutor, e clicar em "Iniciar sessão", deve ser apresentada a página principal desse Condutor, com funcionalidades exclusivas do cargo de Condutor referentes à sua conta.		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.2: Requisito funcional 1.

Requirement #2	Type:	Event/Use Case:
Description: Como Condutor, quero poder encerrar a minha sessão a qualquer momento, para sair da minha conta.		
Rationale: Garantir que um Condutor pode encerrar a sessão, seja para trocar de conta ou somente sair.		
Originator: Introspeção		
Fit Criterion: Um Condutor encerra a sessão em que está e volta para a janela de início de sessão		
Customer Satisfaction: 5		Customer Dissatisfaction: 5
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.3: Requisito funcional 2.

Requirement #3	Type:	Event/Use Case:2
Description: Como novo Condutor, quero poder criar uma conta inserindo o meu nome completo, endereço de e-mail, NIF, número de telemóvel, género e palavra-passe para ter acesso às funcionalidades exclusivas da minha conta.		
Rationale: Manter registo dos Condutores, e assim poder associar as ações a uma certa conta, garantindo a segurança da aplicação.		
Originator: Introspeção		
Fit Criterion: Após o registo, deve ser possível iniciar sessão utilizando o nome de utilizador e palavra-passe introduzidos no ato de registo.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.4: Requisito funcional 3.

Requirement #4	Type: 9	Event/Use Case:3
Description: O Condutor devo poder atualizar o meu perfil pessoal.		
Rationale: A capacidade de atualizar informações do perfil é importante para garantir que os dados do utilizador sejam precisos e estejam atualizados, o que ajuda a manter a integridade do sistema e a fornecer informações mais relevantes.		
Originator: Introspeção		
Fit Criterion: O sistema deve fornecer um formulário de registo de Condutores com campos obrigatórios para o nome do utilizador, endereço de e-mail e senha. O sistema deve permitir que os Condutores atualizem as informações do perfil a qualquer momento.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 23/04/2023		

Tabela 6.5: Requisito funcional 4.

Requirement #5	Type: 9	Event/Use Case:
Description: Como condutor quero poder procurar parques de estacionamento.		
Rationale: A capacidade de pesquisar parques de estacionamento é uma funcionalidade crucial em um sistema de gestão de estacionamento, permitindo aos condutores encontrar facilmente locais disponíveis para estacionar os seus veículos. É igualmente importante que os condutores possam pesquisar por características que esperam que o parque possua, melhorando a experiência do utilizador e reduzindo o tempo necessário para encontrar um local de estacionamento adequado.		
Originator: Introspeção		
Fit Criterion: Os utilizadores devem ser capazes de realizar uma pesquisa de parques de estacionamento com sucesso, utilizando diferentes critérios de pesquisa, como nome, localização, tipo de estacionamento, preço, capacidade e horário de funcionamento.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 23/04/2023		

Tabela 6.6: Requisito funcional 5.

Requirement #6	Type:	Event/Use Case:
Description: Como Condutor, deverei ser capaz de visualizar uma listagem das minhas reservas.		
Rationale: Permite os condutores visualizar todas as reservas que efetuaram.		
Originator: Introspeção		
Fit Criterion: Um Condutor deve ter acesso a uma lista que contém todas as reservas já efetuadas.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 25/04/2023		

Tabela 6.7: Requisito funcional 6.

Requirement #7	Type:	Event/Use Case:4
Description: Como Condutor, deverei poder reservar um lugar de estacionamento por um período de tempo começando num dia e hora selecionados por mim (reserva agendada).		
Rationale: Permite aos condutores, garantir a disponibilidade do lugar durante um certo período de tempo começando num dia e hora selecionados pelo condutor.		
Originator: Introspeção		
Fit Criterion: Um Condutor pode efetuar uma reserva de um lugar, para um certo de estacionamento, dando um dia, uma hora de início e uma hora de fim. Após efetuada a reserva, deve ser possível observar a sua presença na lista de reservas do condutor.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.8: Requisito funcional 7.

Requirement #8	Type:	Event/Use Case:5
Description: Como Condutor, deverei poder reservar um lugar de estacionamento com início imediato, por um período de tempo indeterminado (reserva instantânea).		
Rationale: Permite aos condutores a reserva imediata de um lugar de estacionamento. O facto de não fornecer uma período de tempo é útil para quando o condutor não consegue prever o tempo de permanência do seu veículo no parque. É também útil caso um condutor não se importe de pagar um tempo extra para garantir a disponibilidade do lugar quando chegar ao parque.		
Originator: Introspeção		
Fit Criterion: Um Condutor pode efetuar uma reserva de um lugar, para um certo de estacionamento, com início imediato. Após efetuada a reserva, deve ser possível observar a sua presença na lista de reservas do condutor.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 1/05/2023		

Tabela 6.9: Requisito funcional 8.

Requirement #9	Type:	Event/Use Case:
Description: Como Condutor, quero ser capaz de visualizar a quantidade de lugares ocupados comparando com o total de lugares de um parque de estacionamento.		
Rationale: Permitir aos Condutores ter uma noção da taxa de ocupação do parque.		
Originator: Introspeção		
Fit Criterion: Para todos os parques de estacionamento, seja na página deste, ou numa listagem de parques, devem ser apresentadas ao Condutor, a quantidade de lugares ocupados e o total de lugares do parque de estacionamento.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.10: Requisito funcional 9.

Requirement #10	Type:	Event/Use Case:
Description: Como Condutor, deverei poder efetuar o pagamento de forma digital. O meio de pagamento deve ser um dos seguintes: MB Way ou cartão de crédito.		
Rationale: Aumentar o conforto dos Condutores permitindo efetuar o pagamento sem se terem de deslocar ao guichet de pagamento.		
Originator:		
Fit Criterion: O Condutor deverá conseguir efetuar o pagamento a partir da aplicação.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.11: Requisito funcional 10.

Requirement #11	Type:	Event/Use Case:
Description: Como Condutor, deverei ser notificado quando tiver apenas 15 minutos para finalizar o meu estacionamento pré-pago (referente a reserva agendada)		
Rationale: Relembrar utilizador que o tempo da reserva pré-paga está a acabar.		
Originator:		
Fit Criterion: O Condutor recebe uma notificação quando faltarem 15 minutos para o final do seu estacionamento pré-pago.		
Customer Satisfaction: 3 Customer Dissatisfaction: 3 Priority: Could		
Conflicts: Nenhum		
History: 19/04/2023(criação) 17/06/2023(alteração)		

Tabela 6.12: Requisito funcional 11.

Requirement #12	Type:	Event/Use Case:
Description: Como Condutor, deverei ser notificado quando tiver saído de um parque de estacionamento.		
Rationale: Permitir ao Condutor verificar se o sistema registou corretamente a sua saída do parque de estacionamento.		
Originator:		
Fit Criterion: Após a saída de um Condutor é enviada uma notificação com a confirmação da sua saída.		
Customer Satisfaction: 3	Customer Dissatisfaction: 2	Priority: Could
Conflicts: Nenhum		
History: 19/04/2023(criação) 17/06/2023(alteração)		

Tabela 6.13: Requisito funcional 12.

Requirement #13	Type:	Event/Use Case:
Description: Como Condutor, deverei ser notificado sobre o estado do pagamento.		
Rationale: Permitir ao Condutor saber se o pagamento correu bem ou se deu erro.		
Originator:		
Fit Criterion: Após a finalização do pagamento, deverá ser enviado uma notificação ao utilizador a informar se a transação foi aprovada ou se houve algum erro.		
Customer Satisfaction: 4	Customer Dissatisfaction: 3	Priority: Could
Conflicts: Nenhum		
History: 19/04/2023(criação) 17/06/2023(alteração)		

Tabela 6.14: Requisito funcional 13.

Requirement #14	Type:	Event/Use Case:
Description: Como Condutor, deverei ser notificado quando tiver entrado num parque de estacionamento.		
Rationale: Permitir ao Condutor verificar que o sistema registou corretamente a sua entrada do parque de estacionamento.		
Originator:		
Fit Criterion: Após a entrada do Condutor no parque de estacionamento, é enviada uma notificação com a confirmação da sua entrada.		
Customer Satisfaction: 3	Customer Dissatisfaction: 3	Priority: Could
Conflicts: Nenhum		
History: 19/04/2023(criação) 17/06/2023(alteração)		

Tabela 6.15: Requisito funcional 14.

Requirement #15	Type:	Event/Use Case:
Description: Como Condutor, deverei ser capaz de comparar preços de parques usando o tempo esperado de permanência.		
Rationale: Permitir aos Condutores tomar uma decisão mais económica e consciente sobre qual parque escolher, usando como base o tempo esperado de permanência comparando os preços entre as várias possibilidades apresentadas.		
Originator: Introspeção		
Fit Criterion: Um condutor deve poder selecionar os parques que pretende comparar. Finalmente, após clicar na opção que permite exibir a comparação, deverá surgir uma tabela nas quais é possível comparar os preços dos diversos parques.		
Customer Satisfaction: 4 Customer Dissatisfaction: 2 Priority: Could		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.16: Requisito funcional 15.

Requirement #16	Type:	Event/Use Case:
Description: Como Condutor, deverei poder recarregar mais tempo quando o meu tempo estiver a acabar.		
Rationale: Permitir ao Condutor prolongar a sua reserva se houver disponibilidade para o fazer.		
Originator:		
Fit Criterion: Quando o lugar reservado não tem uma outra reserva agendada logo a seguir, é permitido ao Condutor aumentar a duração da sua reserva.		
Customer Satisfaction: 4 Customer Dissatisfaction: 3 Priority: Could		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.17: Requisito funcional 16.

6.3 Requisitos sobre Programador

Requirement #17	Type:	Event/Use Case:
Description: Como Programador, quero poder iniciar sessão com meu email e palavra-passe.		
Rationale: Para que o Programador tenha acesso às funcionalidades exclusivas de Programador que somente ele pode realizar, contribuindo para a segurança da aplicação.		
Originator: Introspeção		
Fit Criterion: Na página de início de sessão, após introduzir as credenciais (email e palavra-passe) de uma conta de Programador, e clicar em "Iniciar sessão", deve ser apresentada a página principal de Programador, com funcionalidades exclusivas do cargo de Programador.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.18: Requisito funcional 17.

Requirement #18	Type: 9	Event/Use Case:
Description: Como Programador, quero poder encerrar a minha sessão a qualquer momento, para sair da minha conta.		
Rationale: Garantir que um Programador pode encerrar a sessão, seja para trocar de conta ou somente sair.		
Originator: Introspeção		
Fit Criterion: Um Programador encerra a sessão em que está e volta para a janela de início de sessão.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.19: Requisito funcional 18.

Requirement #19	Type: 9	Event/Use Case: 16
Description: Como Programador, deverei ter acesso à informação da utilização da aplicação, nomeadamente, volumes de estacionamento, montantes totais pagos (<i>e mais</i>).		
Rationale: Permite fornecer informações para o Programador sobre o uso global da aplicação, para que este possa tomar decisões sobre a gestão da aplicação.		
Originator: Introspeção		
Fit Criterion: Um Programador autenticado deve ter acesso às diversas métricas globais (i.e. que contabilizam todas as entidades relacionadas com essa métrica) de utilização da aplicação. No caso da métrica "volumes de estacionamento", os valores devem ser apresentados como um somatório dos volumes de estacionamento de todos os parques. A métrica "montantes totais pagos" corresponde ao somatório dos montantes pagos em todos os parques.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.20: Requisito funcional 19.

Requirement #20	Type: 9	Event/Use Case: 15
Description: Como Programador, deverei ser capaz de criar um Gestor de parques, definindo um email, uma palavra-passe e os parques que este deve gerir.		
Rationale: Permite criar contas que possam efetuar a gestão de um ou mais parques de estacionamento.		
Originator: Introspeção		
Fit Criterion: Deve ser possível iniciar sessão na conta de gestor criada com as credenciais introduzidas pelo programador, e verificar a capacidade de gerir os parques especificados.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.21: Requisito funcional 20.

Requirement #21	Type: 9	Event/Use Case: 14
Description: Como Programador, deverei ser capaz de registrar um novo parque de estacionamento, fornecendo informações necessárias como o nome e a localização deste.		
Rationale: Registar novos parques de estacionamento é uma funcionalidade importante porque permite incluir novos locais de estacionamento, alargando as opções de estacionamento que permitem aos condutores tirar proveito das funcionalidades da aplicação.		
Originator: Introspeção		
Fit Criterion: Após o registo do novo parque, deve ser possível verificar a presença deste na listagem de parques registados na aplicação.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.22: Requisito funcional 21.

Requirement #22	Type: 9	Event/Use Case:
Description: Como Programador, deverei ser capaz de eliminar um Gestor de parques.		
Rationale: Permite eliminar contas que já não devem ser utilizadas.		
Originator: Introspeção		
Fit Criterion: Tentar iniciar sessão com as credenciais do gestor eliminado deve resultar em insucesso.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.23: Requisito funcional 22.

Requirement #23	Type: 9	Event/Use Case:
Description: Como Programador, deverei ser capaz de eliminar um parque de estacionamento.		
Rationale: Permite eliminar parques de estacionamento que deixam de estar relacionados com a aplicação.		
Originator: Introspeção		
Fit Criterion: O parque deve deixar de aparecer nas listagens de parques da aplicação.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.24: Requisito funcional 23.

Requirement #24	Type: 9	Event/Use Case: 17
Description: Como Programador, deverei ser capaz de atribuir um determinado parque a um certo gestor.		
Rationale: Permite atribuir, a um gestor, a permissão de gerir um certo parque de estacionamento, evitando a necessidade de criar várias contas, uma por cada parque.		
Originator: Introspeção		
Fit Criterion: Após iniciada sessão na conta do gestor em questão, deve ser possível verificar a capacidade de gerir o parque que lhe foi atribuído.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.25: Requisito funcional 24.

Requirement #25	Type: 9	Event/Use Case:
Description: Como Programador, deverei ser capaz de desatribuir um determinado parque a um certo gestor.		
Rationale: Permite desatribuir, a um gestor, a permissão de gerir um certo parque de estacionamento.		
Originator: Introspeção		
Fit Criterion: Após iniciada sessão na conta do gestor em questão, deve ser possível verificar que já não possui a capacidade de gerir o parque desatribuído.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 20/04/2023		

Tabela 6.26: Requisito funcional 25.

Requirement #26	Type: 9	Event/Use Case: 19
Description: Como Programador, deverei ser capaz de procurar um parque introduzindo apenas o nome deste.		
Rationale: Permite encontrar de forma rápida um parque que necessita de alguma alteração, seja a atribuição de um novo gestor, ou até a sua eliminação.		
Originator: Introspeção		
Fit Criterion: Com a sessão iniciada numa conta de Programador, devo ter acesso a uma barra de procura que me permite introduzir o nome de um parque, e desta forma conseguir encontrá-lo de forma rápida.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 20/04/2023		

Tabela 6.27: Requisito funcional 26.

Requirement #27	Type: 9	Event/Use Case:
Description: Como Programador, deverei ser capaz de procurar um gestor introduzindo apenas o email deste.		
Rationale: Permite encontrar de forma rápida um gestor que necessita de alguma alteração, seja a atribuição de um novo parque, ou até a sua eliminação.		
Originator: Introspeção		
Fit Criterion: Com a sessão iniciada numa conta de Programador, devo ter acesso a uma barra de procura que me permite introduzir o email de um gestor, e desta forma conseguir encontrá-lo de forma rápida.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 20/04/2023		

Tabela 6.28: Requisito funcional 27.

Requirement #28	Type: 9	Event/Use Case: 18
Description: Como Programador, deverei ser capaz de ver a lista de todos os parques registados na aplicação.		
Rationale: Permite encontrar um parque que necessita de alguma alteração, seja para a atribuição de um novo gestor, ou até a sua eliminação.		
Originator: Introspeção		
Fit Criterion: Com a sessão iniciada numa conta de Programador, devo ter acesso à lista de parques registados.		
Customer Satisfaction: 4	Customer Dissatisfaction: 4	Priority: Should
Conflicts: Nenhum		
History: 20/04/2023		

Tabela 6.29: Requisito funcional 28.

Requirement #29	Type: 9	Event/Use Case:
Description: Como Programador, deverei ser capaz de ver a lista de gestores registados na aplicação.		
Rationale: Permite encontrar um gestor que necessita de alguma alteração, seja a atribuição de um novo parque, ou até a sua eliminação.		
Originator: Introspeção		
Fit Criterion: Com a sessão iniciada numa conta de Programador, devo ter acesso à lista de gestores registados.		
Customer Satisfaction: 4	Customer Dissatisfaction: 4	Priority: Should
Conflicts: Nenhum		
History: 20/04/2023		

Tabela 6.30: Requisito funcional 29.

6.4 Requisitos sobre Gestor

Requirement #30	Type: 9	Event/Use Case:
Description: Como Gestor, quero poder iniciar sessão com o meu email e palavra-passe.		
Rationale: Para que o Gestor tenha acesso às funcionalidades exclusivas de Gestor que somente ele pode realizar, contribuindo para a segurança da aplicação.		
Originator: Introspeção		
Fit Criterion: Na página de início de sessão, após introduzir as credenciais (email e palavra-passe) de uma conta de Gestor, e clicar em "Iniciar sessão", deve ser apresentada a página principal desse Gestor, com funcionalidades exclusivas do cargo de Gestor referentes à sua conta.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.31: Requisito funcional 30.

Requirement #31	Type: 9	Event/Use Case:
Description: Como Gestor, quero poder encerrar a minha sessão a qualquer momento.		
Rationale: Garantir que um Gestor pode encerrar a sessão, seja para trocar de conta ou somente sair.		
Originator: Introspeção		
Fit Criterion: Um Gestor encerra a sessão em que está e volta para a janela de início de sessão		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.32: Requisito funcional 31.

Requirement #32	Type: 9	Event/Use Case: 11
Description: Como Gestor, deverei ter acesso a estatísticas sobre volumes de estacionamento dos parques de estacionamento que tenho permissão para gerir.		
Rationale: Permite fornecer informações para o gestor tomar decisões sobre a gestão do parque de estacionamento, como horários de funcionamento, preços, etc.		
Originator: Introspeção		
Fit Criterion: O gestor deve ser capaz de aceder às estatísticas de ocupação dos parques de estacionamento que tem permissão para gerir.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.33: Requisito funcional 32.

Requirement #33	Type: 9	Event/Use Case: 11
Description: Como Gestor, deverei ter acesso a estatísticas sobre montantes totais pagos.		
Rationale: Permite fornecer ao gestor acesso a informações sobre as receitas geradas pelo parque de estacionamento.		
Originator: Introspeção		
Fit Criterion: O gestor deve ser capaz de aceder às estatísticas de montantes totais pagos do parque de estacionamento.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.34: Requisito funcional 33.

Requirement #34	Type: 9	Event/Use Case:
Description: Como Gestor, quero ser capaz de criar contas de Administradores de parques, definindo um email, uma palavra-passe e os parques que este deve administrar. Apenas podem ser atribuídos parques que o próprio gestor tenha permissão de gerir.		
Rationale: Permite criar contas que possam administrar um ou mais parques de estacionamento.		
Originator: Introspeção		
Fit Criterion: Deve ser possível iniciar sessão na conta de administrador criada com as credenciais introduzidas pelo gestor, e verificar a capacidade de administrar os parques especificados.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.35: Requisito funcional 34.

Requirement #35	Type: 9	Event/Use Case: 12
Description: Como Gestor, quero ser capaz de atribuir um parque a uma conta de administrador de parques já existente. Apenas pode ser atribuído um parque que o próprio gestor tenha permissão de gerir.		
Rationale: Permite atribuir, a um administrador, a permissão de administrar um certo parque de estacionamento, evitando a necessidade de criar várias contas, uma por cada parque.		
Originator: Introspeção		
Fit Criterion: Após iniciada sessão na conta do administrador em questão, deve ser possível verificar a capacidade de administrar o parque que lhe foi atribuído.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.36: Requisito funcional 35.

Requirement #36	Type: 9	Event/Use Case:
Description: Como Gestor, quero ser capaz de desatribuir a permissão de administrar um determinado parque a uma conta de administrador.		
Rationale: Permite desatribuir, a um administrador, a permissão de administrar um certo parque de estacionamento.		
Originator: Introspeção		
Fit Criterion: Após iniciada sessão na conta do administrador em questão, deve ser possível verificar que já não possui a capacidade de administrar o parque que lhe foi desatribuído.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.37: Requisito funcional 36.

Requirement #37	Type: 9	Event/Use Case:
Description: Como Gestor, deverei ser capaz de ver a lista de parques que tenho permissão para gerir.		
Rationale: Permite encontrar o parque que o Gestor pretende gerir.		
Originator: Introspeção		
Fit Criterion: Com a sessão iniciada numa conta de Gestor, devo ter acesso à lista de parques que esta conta tem permissão para gerir.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 20/04/2023		

Tabela 6.38: Requisito funcional 37.

Requirement #38	Type: 9	Event/Use Case:13
Description: Como Gestor, quero ser capaz de marcar o parque de estacionamento como aberto ou fechado.		
Rationale: Permite que um gestor defina o estado em que um parque de estacionamento se encontra. Se um parque precisa de entrar em manutenção, o gestor pode declarar o parque como fechado, impedindo a realização de reservas por parte dos condutores.		
Originator: Introspeção		
Fit Criterion: O estado do parque em questão deve ser mudado para o estado escolhido pelo gestor. No caso de o estado ser "Aberto", a funcionalidade de reservas deve passar a ser permitida. No caso de o estado ser "Fechado", a funcionalidade de reservas deve ser desativada.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.39: Requisito funcional 38.

Requirement #39	Type: 9	Event/Use Case:
Description: Como Gestor, quero ser capaz de gerir as tabelas de preços dos parques de estacionamento que eu tenho permissão para gerir.		
Rationale: Permite a definição personalizada dos preços das reservas.		
Originator: Introspeção		
Fit Criterion: Numa conta de gestor com sessão iniciada, e após selecionar um dos parques que posso gerir, devo poder verificar a opção de gerir a tabela de preços desse parque. Ao fim de feitas essas alterações, estas devem poder ser visualizadas pelos condutores que acedam à página do parque de estacionamento.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 25/04/2023		

Tabela 6.40: Requisito funcional 39.

Requirement #40	Type: 9	Event/Use Case:
Description: Como Gestor, quero ser capaz de gerir o horário de funcionamento dos parque de estacionamento que eu tenho permissão para gerir.		
Rationale: Permite a definição personalizada do horário de funcionamento dos parques de estacionamento, de forma a limitar as horas em que é possível efetuar reservas. As reservas agendadas podem ser feitas a qualquer hora, no entanto o intervalo reservado deve constar dentro do horário de funcionamento definido.		
Originator: Introspeção		
Fit Criterion: Numa conta de gestor com sessão iniciada, e após selecionar um dos parques que posso gerir, devo poder verificar a opção de gerir o horário de funcionamento desse parque. Ao fim de feitas essas alterações, estas devem poder ser visualizadas pelos condutores que acedam à página do parque de estacionamento.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 25/04/2023		

Tabela 6.41: Requisito funcional 40.

6.5 Requisitos sobre Administrador

Requirement #41	Type: 9	Event/Use Case:
Description: Como Administrador, quero poder iniciar sessão com o meu email e palavra-passe.		
Rationale: Para que o administrador tenha acesso às funcionalidades exclusivas de administração que somente ele pode realizar, contribuindo para a segurança da aplicação.		
Originator: Introspeção		
Fit Criterion: Na página de início de sessão, após introduzir as credenciais (email e palavra-passe) de uma conta de Administrador, e clicar em "Iniciar sessão", deve ser apresentada a página principal dess Administrador, com funcionalidades exclusivas do cargo de Administrador referentes à sua conta.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.42: Requisito funcional 41.

Requirement #42	Type: 9	Event/Use Case:
Description: Como administrador, quero poder encerrar a minha sessão a qualquer momento.		
Rationale: Garantir que um administrador pode encerrar a sessão, seja para trocar de conta ou somente sair.		
Originator: Introspeção		
Fit Criterion: Um administrador encerra a sessão em que está e volta para a janela de início de sessão		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.43: Requisito funcional 42.

Requirement #43	Type: 9	Event/Use Case:6
Description: Como Administrador, devo poder adicionar um lugar ao parque, seja o lugar para reservas instantâneas ou agendadas.		
Rationale: Permitir incrementar o número de lugares que estão disponíveis para estacionamento no parque de estacionamento.		
Originator: Introspeção		
Fit Criterion: No caso dos lugares de reservas instantâneas, deve ser possível verificar o aumento de 1 unidade no número total de lugares de estacionamento desse tipo de reservas. Se o lugar for para reservas agendadas, deve ser possível gerir/reservar este lugar especificamente.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.44: Requisito funcional 43.

Requirement #44	Type: 9	Event/Use Case: 7
Description: Como Administrador, devo poder remover um lugar do parque, seja o lugar para reservas instantâneas ou agendadas.		
Rationale: Permitir decrementar o número de lugares que estão disponíveis para estacionamento no parque de estacionamento.		
Originator: Introspeção		
Fit Criterion: No caso dos lugares de reservas instantâneas, deve ser possível verificar a diminuição em 1 unidade no número total de lugares de estacionamento de reservas instantâneas. Se o lugar for para reservas agendadas, deve desaparecer, deixando de ser possível gerir/reservar esse lugar.		
Customer Satisfaction: 5	Customer Dissatisfaction: 5	Priority: Must
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.45: Requisito funcional 44.

Requirement #45	Type: 9	Event/Use Case: 8
Description: Como Administrador, devo poder ativar um lugar de reservas agendadas.		
Rationale: Permitir que um lugar de estacionamento possa ser reservado por um condutor.		
Originator: Introspeção		
Fit Criterion: O lugar deve poder ser reservado por um condutor.		
Customer Satisfaction: 2	Customer Dissatisfaction: 1	Priority: Could
Conflicts: Nenhum		
History: 19/04/2023(criado) 17/06/2023(alterado)		

Tabela 6.46: Requisito funcional 45.

Requirement #46	Type: 9	Event/Use Case: 9
Description: Como Administrador, devo poder desativar um lugar de reservas agendadas.		
Rationale: Impedir que um lugar de estacionamento seja reservado por um condutor.		
Originator: Introspeção		
Fit Criterion: O lugar deixa de poder ser reservado por qualquer condutor.		
Customer Satisfaction: 2	Customer Dissatisfaction: 1	Priority: Could
Conflicts: Nenhum		
History: 19/04/2023(criação) 17/06/2023(alterado)		

Tabela 6.47: Requisito funcional 46.

Requirement #47	Type: 9	Event/Use Case: 10
Description: Como Administrador, quero poder visualizar uma reserva através de um identificador desta.		
Rationale: Para permitir que um condutor entre no parque de estacionamento, caso a reserva esteja ativa. Ou para permitir que este saia, caso o montante devido estiver pago.		
Originator: Introspeção		
Fit Criterion: O administrador consegue ver uma reserva caso o identificador fornecido pelo condutor seja válido.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.48: Requisito funcional 47.

Requirement #48	Type: 9	Event/Use Case:
Description: Como administrador, quero poder associar uma matrícula a uma reserva ativa.		
Rationale: Para poder criar uma associação entre a reserva e um veículo, permitindo posteriormente identificar o condutor que estacionou o veículo.		
Originator: Introspeção		
Fit Criterion: Uma matrícula fica associada à reserva.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 19/04/2023		

Tabela 6.49: Requisito funcional 48.

6.6 Requisitos de Sistema

Requirement #49	Type: 9	Event/Use Case:
Description: Uma reserva tem 4 estados: pendente pagamento, agendada, ocupada, concluída.		
Rationale: Este requisito é importante para garantir que o sistema possa gerir efetivamente o processo de reserva, garantindo que a reserva seja atualizada automaticamente à medida que ela progride pelos diferentes estados. Também é importante garantir que uma reserva só possa ser agendada depois de ter sido paga, para evitar reservas em duplicado ou conflitos de horários.		
Originator: Introspeção		
Fit Criterion: O sistema deve atualizar o estado da reserva automaticamente à medida que a reserva progride pelos quatro estados: pendente pagamento, agendada, ocupada e concluída. O sistema não deve permitir que um usuário agende uma reserva que ainda não tenha sido paga.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 23/04/2023		

Tabela 6.50: Requisito funcional 49.

Requirement #50	Type: 9	Event/Use Case:
Description: O sistema deverá permitir que os pagamentos das reservas sejam efetuados utilizando um dos seguintes meios de pagamento: Cartão de crédito, Multibanco ou MB Way.		
Rationale: Este requisito define os métodos de pagamento que o cliente pretende que sejam usados para os pagamentos das reservas.		
Originator: Introspeção		
Fit Criterion: Aquando do pagamento de uma reserva, os condutores devem poder verificar a possibilidade de efetuar o pagamento com qualquer um destes métodos de pagamento.		
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must		
Conflicts: Nenhum		
History: 23/04/2023		

Tabela 6.51: Requisito funcional 50.

7. Requisitos Não Funcionais

7.1 Appearance

Requirement #1	Type: 10
Description: A aparência da aplicação deve transmitir confiança aos utilizadores.	
Rationale: É vontade da empresa que os seus utilizadores se sintam confiantes com o uso da aplicação.	
Originator: Introspeção	
Fit Criterion: Para uma amostra com utilizadores representativos de várias faixas etárias, 70 por cento dos inquiridos devem concordar que se sentem confiantes com o uso da aplicação, logo após o seu primeiro contato com ela.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.1: Requisito não funcional 1.

Requirement #2	Type: 10
Description: A aplicação deve ser facilmente reconhecida e distingível de outras aplicações concorrentes.	
Rationale: A aplicação deverá ser original e distinta para que o utilizador o consiga separar de outras aplicações semelhantes.	
Originator: Introspeção	
Fit Criterion: A aplicação deve ter uma identidade visual única e memorável, com um logotipo distintivo e cores que se destacam da concorrência. Além disso, a aplicação deve ter um nome exclusivo e fácil de lembrar que reflete os principais recursos e benefícios da aplicação.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.2: Requisito não funcional 2.

Requirement #3	Type: 10
Description: O sistema deve ser responsivo e adaptável, garantindo que sua aparência seja otimizada para diferentes tamanhos de tela e dispositivos, incluindo desktops, tablets e smartphones.	
Rationale: É essencial que o sistema seja responsivo e adaptável para garantir que sua aparência e layout sejam otimizados para diferentes tamanhos de tela e dispositivos, mantendo sua usabilidade e funcionalidade em todos os dispositivos.	
Originator: Introspeção	
Fit Criterion: O sistema deve ser capaz de ajustar sua aparência e layout de acordo com o tamanho da tela do dispositivo usado pelo usuário, mantendo sua usabilidade e funcionalidade em diferentes dispositivos, incluindo desktops, tablets e smartphones.	
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must	
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.3: Requisito não funcional 3.

7.2 Usability

Requirement #4	Type: 11
Description: A aplicação deve ter recursos de pesquisa e filtragem para facilitar a localização de informações específicas.	
Rationale: É desejável que a aplicação permita ao utilizador pesquisar e/ou filtrar pelas características mais relevantes às varias listas de informação que vão ser apresentadas de forma a que obtenha os resultados pretendidos mais eficazmente e num espaço de tempo mais curto.	
Originator: Introspeção	
Fit Criterion: O sistema deve permitir que o usuário pesquise e filtre informações usando palavras-chave, categorias e filtros personalizados. O resultado da pesquisa deve ser preciso e relevante para as palavras-chave ou filtros usados.	
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must	
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.4: Requisito não funcional 4.

Requirement #5	Type: 11
Description: A aplicação deve ser capaz de ser utilizada por utilizadores que não receberam treino antes de a utilizar.	
Rationale: É desejável que a aplicação seja simples de compreender de forma a que qualquer utilizador a possa utilizar mesmo que seja o seu primeiro contacto com ela.	
Originator: Introspeção	
Fit Criterion: 80 por cento de um painel com utilizadores representativos de várias faixas etárias, devem ser capazes de completar, com sucesso, uma reserva de um lugar para um período de 3h no dia seguinte, demorando menos de 10 minutos e sem ajuda externa.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.5: Requisito não funcional 5.

Requirement #6	Type: 11
Description: A aplicação deve fazer o usuário querer utilizá-la.	
Rationale: É desejável que esta aplicação se revele interessante para os utilizadores e que eles continuem a utiliza-la.	
Originator: Introspeção	
Fit Criterion: Uma pesquisa anônima deve demonstrar que 90 por cento dos utilizadores estão a utilizar regularmente a aplicação após um período de três semanas de familiarização.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.6: Requisito não funcional 6.

Requirement #7	Type: 11
Description: A aplicação deve usar palavras e símbolos que possam ser facilmente compreendidos.	
Rationale: Usar palavras e símbolos que os utilizadores reconheçam por estarem associadas a ações semelhantes em outras aplicações, ajuda a que estes utilizadores cometam menos erros e achem a aplicação mais intuitiva e simples de usar.	
Originator: Introspeção	
Fit Criterion: A aplicação deve usar terminologia clara e concisa, com palavras e símbolos que sejam facilmente compreendidos pelos utilizadores, mesmo que não tenham experiência técnica. O sistema deve evitar palavras técnicas e termos ambíguos que possam confundir os utilizadores.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.7: Requisito não funcional 7.

Requirement #8	Type: 11
Description: A aplicação deve ser possível de usar por pessoas daltónicas.	
Rationale: A aplicação deverá ter em conta as dificuldade de distinguir cores característica das pessoas com daltonismo e adaptar-se para que mesmo essas pessoas consigam usar a aplicação.	
Originator: Introspeção	
Fit Criterion: Usando como amostra um conjunto de utilizadores representativos dos vários graus de daltonismo, deverá ser possível constatar que mais de 95 por cento desses utilizadores conseguiram utilizar a aplicação sem nenhum impedimento provocado pelo daltonismo.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.8: Requisito não funcional 8.

Requirement #9	Type: 11
Description: O sistema deve incluir uma aplicação móvel dedicada para os Condutores que poderá ser transferida e instalada nos seus dispositivos móveis a partir das principais lojas de aplicações móveis(App Store e Play Store).	
Rationale: A disponibilização de uma aplicação móvel para os Condutores vai permitir aumentar a acessibilidade da mesma, uma vez que, a maioria dos utilizadores da aplicação vão ser Condutores, que quando quiserem utilizar a aplicação não vão ter acesso a computadores.	
Originator: Introspeção	
Fit Criterion: No final do desenvolvimento do sistema vamos ter disponível uma aplicação móvel compatível com os sistemas operativos Android ou IOS.	
Customer Satisfaction: 4	Customer Dissatisfaction: 4
Conflicts: Nenhum	
History: 21/04/2023(criado) 17/06/2023(alterado)	

Tabela 7.9: Requisito não funcional 9.

7.3 Performance

Requirement #10	Type: 12
Description: Ações que não acedam à base de dados devem ser realizadas em tempo real.	
Rationale: Vital para uma boa experiência para o utilizador na aplicação.	
Originator: Introspeção	
Fit Criterion: Para ações que não acedam à base de dados, o sistema deve responder na casa dos décimos de segundo.	
Customer Satisfaction: 4	Customer Dissatisfaction: 4
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.10: Requisito não funcional 10.

Requirement #11	Type: 12
Description: O sistema deve estar funcional a maior parte do tempo.	
Rationale: Essencial para o funcionamento da aplicação. Garantir a possibilidade de Condutor aceder a qualquer momento.	
Originator: Introspeção	
Fit Criterion: A aplicação deve estar disponível para uso 97 por cento do tempo.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.11: Requisito não funcional 11.

Requirement #12	Type: 12
Description: A aplicação deve ser capaz de processar até 10.000 clientes simultaneamente.	
Rationale: Garantir que o sistema suporta a carga esperada.	
Originator: Introspeção	
Fit Criterion: O sistema consegue funcionar normalmente mesmo em situações em que estejam até 10000 utilizadores a aceder simultaneamente.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.12: Requisito não funcional 12.

Requirement #13	Type: 12
Description: O sistema deve ser capaz de guardar a informação de, no mínimo, 200.000 utilizadores.	
Rationale: Garantir a expansão da aplicação. Aumento do alcance do produto.	
Originator: Introspeção	
Fit Criterion: O sistema não tem problemas em atualizar a informação de cada utilizador em situações em que estejam até 200.000 utilizadores registados.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.13: Requisito não funcional 13.

7.4 Operational

Requirement #14	Type: 13
Description: A aplicação deverá ser capaz de funcionar nos browsers Google Chrome, Microsoft Edge, Opera e Brave.	
Rationale:	
Originator: Introspeção	
Fit Criterion: Sistema está preparado para continuar o seu normal funcionamento em qualquer dos 4 browsers.	
Customer Satisfaction: 2	Customer Dissatisfaction: 4
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.14: Requisito não funcional 14.

7.5 Maintenance and Support

Requirement #15	Type: 14
Description: A aplicação deverá ser projetada e desenvolvida de forma a poder ser acedida via web.	
Rationale: Ao garantir que a aplicação pode ser acedida via web, a equipa de desenvolvimento maximiza o seu alcance da aplicação e torná-a acessível a uma gama mais ampla de utilizadores.	
Originator: Introspeção	
Fit Criterion: A aplicação deve suportar a sua execução nos principais <i>web browsers</i> e não deve depender de quaisquer tecnologias proprietárias ou específicas da plataforma.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.15: Requisito não funcional 15.

7.6 Security

Requirement #16	Type: 15
Description: O sistema deve ter validações de entrada de dados para impedir que dados incorretos sejam inseridos no sistema.	
Rationale: É essencial que o sistema tenha mecanismos de validação de entrada de dados para garantir que apenas dados corretos e válidos sejam inseridos no sistema. Isso ajuda a prevenir erros e inconsistências nos dados, o que pode afetar negativamente a usabilidade do sistema e a tomada de decisão.	
Originator: Introspeção	
Fit Criterion: O sistema deve verificar se os dados atendem aos requisitos específicos, como comprimento mínimo e máximo, caracteres permitidos e formatação adequada. O sistema deve fornecer <i>feedback</i> instantâneo e claro em caso de entrada de dados incorreta ou incompleta, com sugestões para corrigir o erro.	
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must	
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.16: Requisito não funcional 16.

Requirement #17	Type: 15
Description: O sistema deve ser capaz de proteger contra o abuso intencional por parte de um utilizador.	
Rationale: É importante proteger o sistema contra o abuso intencional por parte de usuários para garantir a segurança e a integridade dos dados e informações armazenados no sistema, bem como a experiência positiva dos usuários ao utilizar o sistema.	
Originator: Introspeção	
Fit Criterion: O sistema deve ser capaz de detectar e impedir atividades maliciosas, como o uso de linguagem ofensiva, tentativas de fraude, tentativas de invasão de contas de outros usuários, entre outros.	
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must	
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.17: Requisito não funcional 17.

Requirement #18	Type: 15
Description: A aplicação deve proteger informações particulares em concordância com leis de privacidade relevantes e, à política de informações da organização.	
Rationale: Cumprir, simultaneamente, as leis de privacidade de dados e as políticas de informações impostas pela empresa que encomendou a aplicação.	
Originator: Introspeção	
Fit Criterion: A aplicação respeita as leis de privacidade de dados e as políticas de informação da empresa.	
Customer Satisfaction: 5 Customer Dissatisfaction: 5 Priority: Must	
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.18: Requisito não funcional 18.

7.7 Legal

Requirement #19	Type: 17
Description: O uso da informação pessoal, de todos os intervenientes da aplicação, deve ser implementado pelo sistema de forma a cumprir o “Ato de Proteção de Dados”.	
Rationale: Cumprimos a legislação do país onde o nosso sistema vai funcionar.	
Originator: Introspeção	
Fit Criterion: A aplicação deverá estar de acordo com as normas estabelecidas pelo “Ato de Proteção de Dados” e foi aprovada pelas autoridades competentes.	
Customer Satisfaction: 5	Customer Dissatisfaction: 5
Conflicts: Nenhum	
History: 21/04/2023	

Tabela 7.19: Requisito não funcional 19.

8. Arquitetura da Solução

8.1 Diagramas Complementares da Model Driven Architecture

Após a definição do modelo de domínio (CIM), *use cases* e dos requisitos (funcionais e não funcionais) da aplicação, estamos em condições de passar às próximas fases do ciclo de vida proposto pela *Model Driven Architecture (MDA)*. Analisando os "documentos" referidos anteriormente, passamos à conceção de um modelo *PIM (Platform Independent Model)*. Neste modelo, focamo-nos em descrever os componentes (definidos como *packages*) da nossa aplicação, informando da lógica de negócio que devia ser implementada por estes, sem referir qualquer tipo de tecnologia. Nesta fase, tentamos estruturar a aplicação de forma a que os componentes tivessem o mínimo de dependências entre eles, optando por uma estrutura hierárquica, em que faríamos o refinamento dos métodos ao integrar os novos componentes, e dessa forma criar funcionalidades mais enriquecidas. Tentamos, também, encontrar maneiras de aplicar padrões de conceção que permitissem a expansibilidade dos componentes, e da aplicação como um todo. Seguido do modelo *PIM*, foi desenvolvido o modelo *PSM (Platform Specific Model)*, que consistiu na transformação do modelo PIM, num modelo mais refinado, introduzindo informações sobre as tecnologias que seriam utilizadas na implementação da aplicação. Neste modelo, incluímos informações como o estereótipos das classes, de forma a declarar as classes que seriam *Entity Beans* e *Session Beans*. Para os *Entity Beans*, definimos quais as chaves principais, e como é que as relações seriam criadas entre os diversos *Entity Beans*. Para os *Session Beans* incluímos informações relativas à capacidade de guardar estado, i.e., se devem ser *stateful* ou *stateless*. Com tudo isto, definimos a base da arquitetura da nossa aplicação. No entanto, para complementar a informação presente no *PSM*, foi gerado automaticamente o diagrama de relações entre entidades (*ERD - Entity Relationship Diagram*), que teve de ser ligeiramente adequado para representar, de forma exata, o pretendido pela nossa equipa de desenvolvimento. Finalmente, recorrendo aos mecanismos de geração de código do *VisualParadigm*, baseados nos diagramas *PSM* e *ERD*, e que foram introduzidos nos tutoriais da unidade curricular de *Arquiteturas Aplicacionais*, foi possível dar um grande salto na implementação da aplicação.

É necessário realçar, que o *PIM* não está numa versão tão atualizada como o *PSM*. Durante a fase de concepção do *PSM*, e mesmo durante a implementação, notou-se a falta de alguns métodos, ou até que uma certa estrutura não seria a ideal. Muitas destas falhas foram sendo atualizadas no *PSM*, mas o *PIM* acabou por ser um pouco negligenciado.

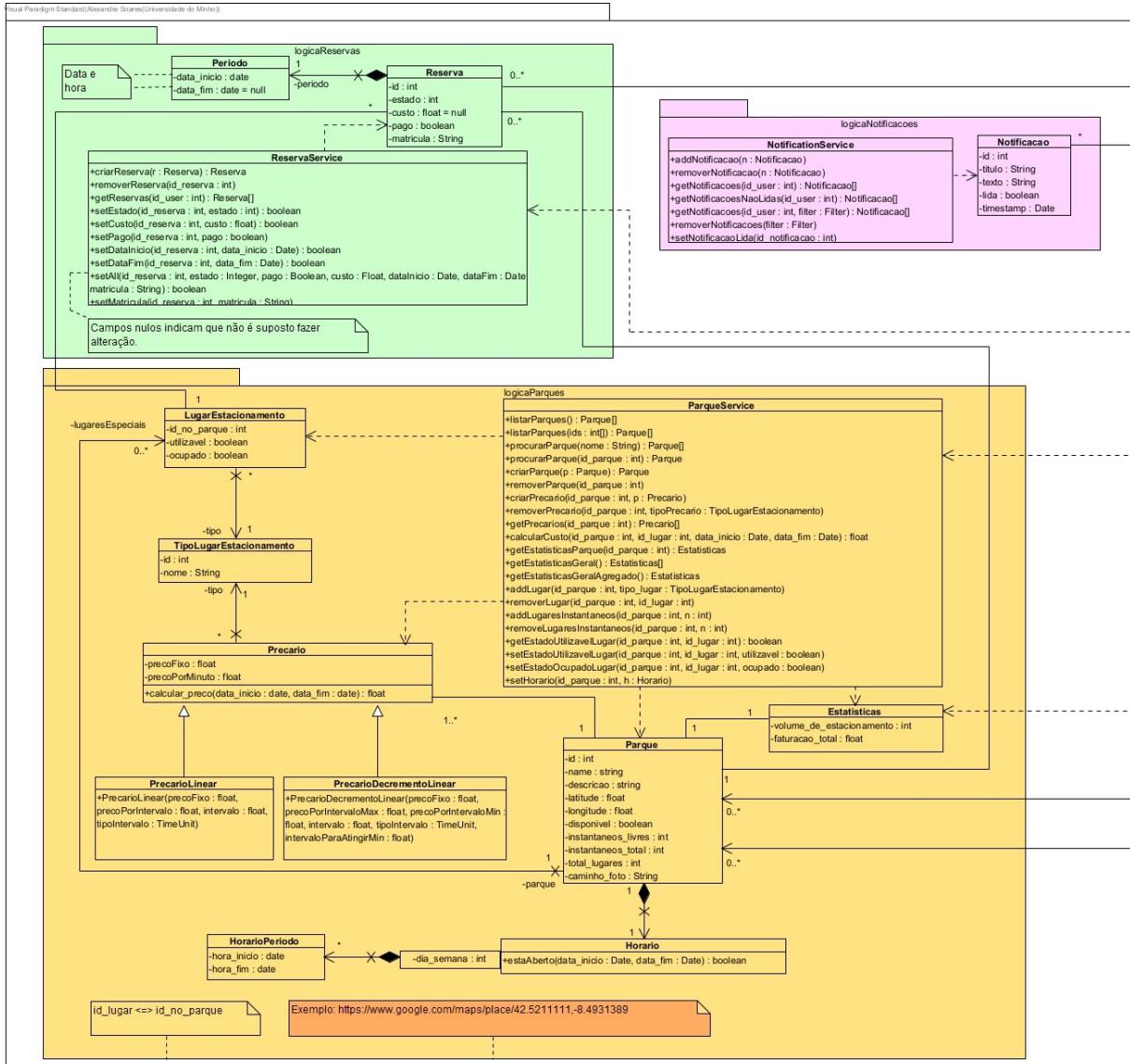


Figura 8.1: PIM com lógica dos parques, reservas e notificações.

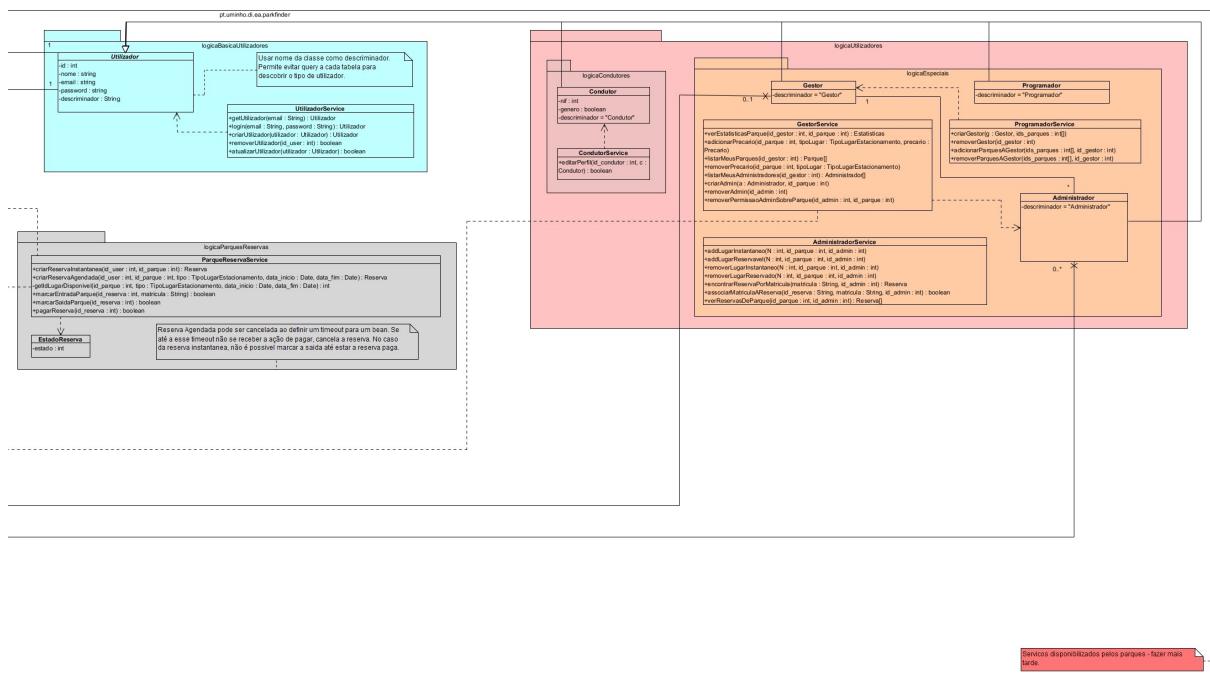


Figura 8.2: PIM com lógica dos utilizadores.

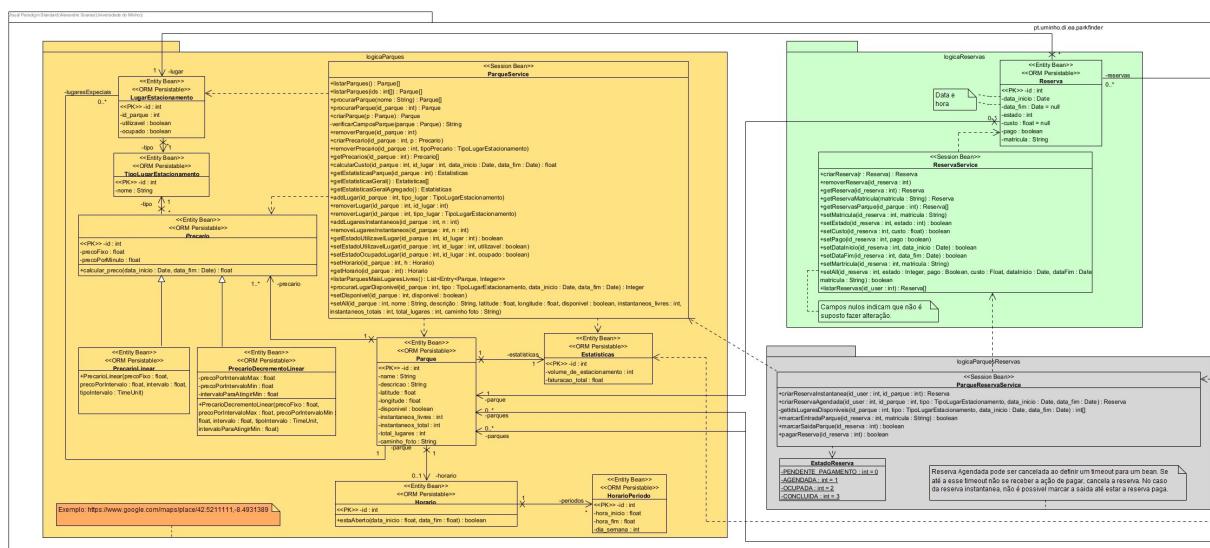


Figura 8.3: PSM com lógica dos parques e das reservas.

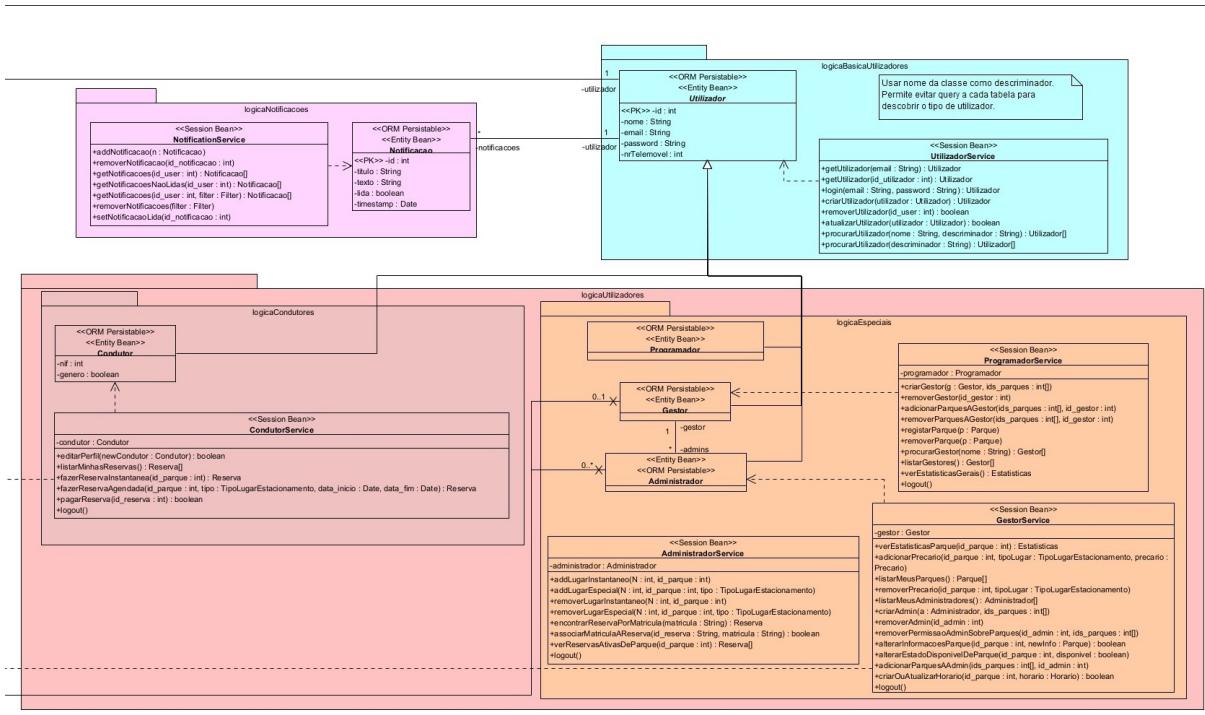


Figura 8.4: PSM com lógica dos utilizadores e das notificações.

Como pode ser visto no diagrama PSM apresentado acima, o grupo decidiu dividir a lógica da aplicação por *packages*. O objetivo desta divisão é a separação de responsabilidades entre cada um dos grupos de classes identificadas no diagrama, promovendo a modularidade, e a evolução independente dos componentes. Como já foi referido anteriormente, tentamos criar uma arquitetura, com um esquema hierárquico, optando pela refinação das funcionalidades, ou criação de novas funcionalidades mais complexas, através da composição de componentes, de forma progressiva. O diagrama e, consequentemente, a aplicação vão possuir 7 *packages* principais, cada um com pelo menos uma classe *Service*, que implementa uma interface, em que o objetivo é permitir uma integração facilitada das funcionalidades fornecidas por um *package*, escondendo a sua complexidade. Resumindo, para cada *package* foi utilizado o padrão de conceção *Facade*.

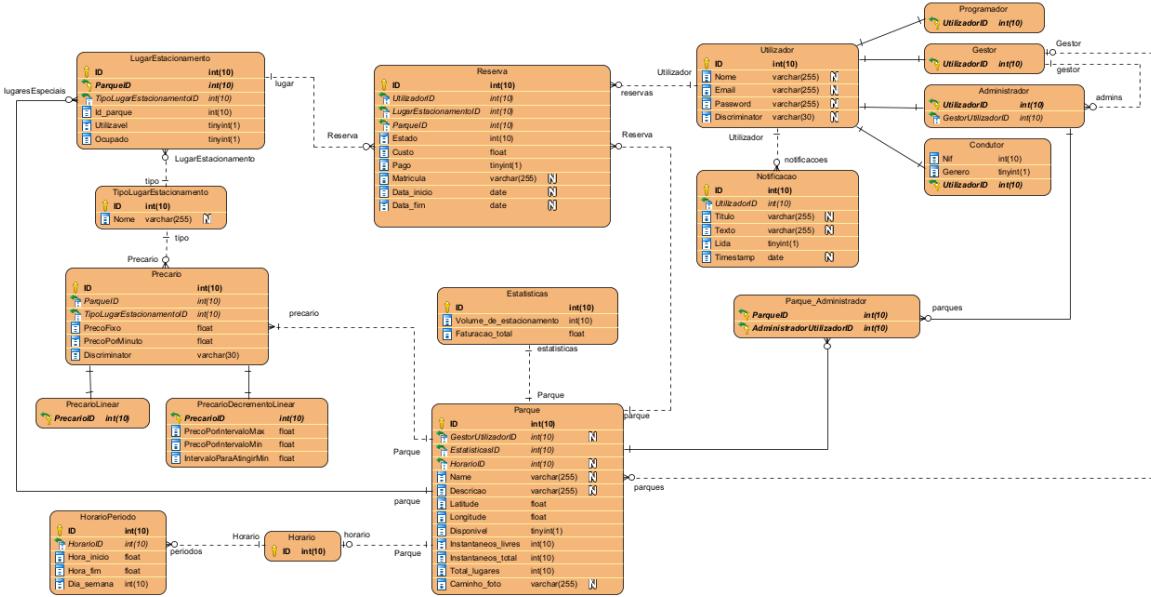


Figura 8.5: Entity Relationship Diagram

8.2 Design Patterns Usados na Solução

Nesta secção, vamos referir que *design patterns* utilizados para desenvolver a solução.

8.2.1 Facade

A utilização do padrão estrutural Facade é bastante evidente no diagrama PSM da secção 8.1, uma vez que, decidimos subdividir o diagrama PSM em *sub-packages* para dividir a lógica das diferentes áreas da nossa aplicação, descartando a necessidade de outros *sub-packages* conhecerem a complexidade por trás da implementação destes. É quase natural utilizarmos estes *design patterns* para, por um lado, apenas expormos para os outros *sub-packages* da aplicação e, por outro lado, para também agruparmos toda as funcionalidades de determinado *sub-package* numa classe só, para facilitar a integração das funcionalidades. Cada *sub-package* possui uma ou mais classes *Service* que foi o nome atribuído às classes que são, na realidade, *Facades*.

8.2.2 Template

O padrão comportamental *Template* é utilizado no *sub-package* que contém a lógica dos parques. Mais concretamente, na parte relativa aos precários dos parques. Foi criada uma classe abstrata *Precário*, que possui um método abstrato - ou seja, a ser implementado pelas subclasses - designado por '*calcular_preco*'. Este método permite compatibilizar os parques com qualquer tipo de precário. As implementações dos diversos tipos de precários, podem divergir significativamente umas das outras, a nível de atributos necessários, complexidade da lógica, etc. Mas mantendo o núcleo que é o cálculo do custo de uma reserva. Dito isto, as subclasses da classe *Precário*, podem ser permutadas, umas pelas outras, facilmente, permitindo uma maior liberdade na definição de um precário para um parque.

8.2.3 Observer

O padrão comportamental *Observer* não foi implementado na solução que foi entregue à equipa docente, mas vamos referir a forma de como estávamos a pensar utilizá-lo na solução. Este padrão iria ser utilizado na parte relacionada com as notificações. Parte que não se apresenta

funcional na solução apresentada. Tendo em mente, uma arquitetura distribuída, com múltiplos servidores aplicacionais, o objetivo seria utilizar uma biblioteca como o *ZeroMQ* ou *RabbitMQ*, para desenvolver um sistema de notificações por eventos, através de uma topologia baseada no padrão *PUBLISH-SUBSCRIBE*. Os *brokers*, na topologia final, seriam encarregues de gerir as subscrições de notificações e por encaminhar as notificações para os servidores que estão interessados nelas. Agora, passando à implementação, do padrão *Observer* concretamente, i.e., dentro de cada instância da aplicação. Aquando de uma operação de início de sessão, um *Bean* associado à sessão do utilizador, chamemos-lhe *UtilizadorBean*, seria responsável por ficar subscrever o *Bean* responsável por receber as notificações (assíncronamente) dos outros servidores, que designaremos por *NotificationBean*. Este último *Bean*, após receber uma notificação, faria a publicação desta notificação, enviando esta para todos os *Beans* que subscreveram a notificação, por exemplo, através de um tópico. Um *UtilizadorBean*, ao receber a notificação, ficaria então encarregue de executar alguma lógica com esta, nomeadamente, persistir a notificação, e fazer o *push* desta para o cliente que está associado à instância da aplicação (assumimos mecanismos para garantir *server affinity*), por exemplo através de mecanismos que recorrem a sockets unidireccionais, como é o caso de *server-sent events*.

8.3 Arquitetura Final da Aplicação

Uma vez que seguimos a abordagem *Model Driven Architecture* para desenvolver a arquitetura da lógica de negócio da nossa aplicação, o grupo fez um esforço para desenvolver um diagrama PSM de forma cuidada para minimizar o tempo gasto a preparar o código Java gerado com base neste diagrama.

Para além dos elementos referidos no diagrama tivemos de adicionar uma camada extra para integrarmos a lógica de negócio com a camada de apresentação da aplicação. As classes desta camada pertencem a um novo módulo denominado API. As classes API do nosso sistema podem-se agrupar em três partes: o conjunto de API's dos utilizadores autenticados(condutor, administrador, gestor, programador), a API para utilizadores não autenticados(utilizador) e a API dos parques. No primeiro conjunto cada API corresponde às permissões de cada utilizador, ou seja, um condutor só consegue usar métodos da classe 'CondutorAPI'. O segundo conjunto permite aos utilizadores não autenticados terem a possibilidade de se autenticar utilizando a API da classe 'UtilizadoresAPI'. No último conjunto, a ParqueAPI é a classe com os métodos que mostram informação relativa aos parques e está disponível para utilizadores autenticados e não autenticados.

As APIs foram desenvolvidas utilizando a tecnologia fornecida pelo Spring MVC, que permite desenvolver *RESTful Web services*. Dado que este tipo de serviços Web, não fornece uma descrição dos serviços, como é conseguido com o *WSDL*, a equipa incluiu no projeto uma dependência que permite gerar uma descrição dos serviços, facilitando assim, o trabalho de quem pretende integrar a API num outro projeto, ou como no nosso caso, para ajudar na integração com a camada de apresentação. Esta dependência é designada de SpringDoc OpenAPI, também designada, por Swagger. Após lançar a aplicação, é possível aceder a esta descrição, através do seguinte URL "*endpoint/swagger-ui.html*".

Figura 8.6: Exemplo do método de login no Swagger

8.4 Instalação/Deployment

Infelizmente, não foi possível fazer um *deployment* das várias camadas da aplicação (de dados, de lógica e de apresentação). Idealmente, teríamos feito esta instalação, por exemplo, na Google Cloud. Configurando um *load balancer*, para distribuir os clientes, pelos diversos servidores aplicacionais. Com *server affinity*, i.e., tentando que os clientes mantenham a conexão com o servidor aplicacional com o qual se ligou inicialmente, de forma a fazer uma melhor gestão dos *stateful session beans* presentes em cada instância. Também seria ideal, a implementação de uma *cache*, nomeadamente, para informações constantes sobre os parques, passando a carga imposta à camada de dados, a ser essencialmente sobre a disponibilidade deste, sobre o número de lugares livres, etc. Foram utilizados repositórios baseados na tecnologia *Jpa* (*JpaRepository*), no entanto, achamos que a configuração de um *Session Factory* com *Hibernate*, teria sido algo mais vantajoso para a escalabilidade da aplicação, como foi abordado nos tutoriais. Como já foi dado a entender, esta aplicação foi desenvolvida utilizando a *framework Spring*. A *framework* fornece alternativas para *stateful/stateless session beans* da tecnologia *EJB*, são os designados *session scoped beans* como alternativa para os *stateful session beans*, e os *singleton beans* com alternativa aos *stateless beans*. Os singleton beans foram utilizados para serviços muito utilizados e que não necessitam de uma sessão, como é o caso do serviço dos parques, em que utilizadores não autenticados podem utilizar o serviço. No caso de ser necessário autenticação, foram criados session scoped beans, para impedir que fossem realizadas operações sem que o utilizador tivesse essa permissão, e de forma a poder manter estado conversacional, evitando enviar constantemente os mesmos dados pela rede. Uma possível otimização que deveríamos ter feito, é relativamente a índices das queries à base de dados e possíveis vistas materializadas, etc, para aumentar a taxa de atendimento às interrogações feitas.

9. Avaliação de interface e princípios de usabilidade

No desenvolvimento da aplicação, ponderamos sobre os princípios de usabilidade e de que melhor forma poderiam ser cumpridos na nossa aplicação, chegando à conclusão que conseguimos garantir os seguintes:

- Capacidade de aprendizagem:
 - **Previsibilidade** - Com o uso de padrões de concepção e com senso comum esperado de um comum utilizador, testamos e concluímos que é possível prever o efeito que o cumprimento das tarefas tem na interface.
 - **Synthesizability (of mental model)**, em todas as páginas asseguramos que quando são feitas tarefas que alteram alguma variável revelante ao utilizador, este consegue ver o seu efeito na respetiva interface.
 - **Generabilidade** - Na criação da aplicação tivemos a noção de fazer as tarefas de cariz parecido, o mais semelhante possível, como por exemplo na edição de informações pessoais, garantindo assim a generabilidade da aplicação.
 - **Familiaridade** - para criar uma interface onde conhecimento passado torna intuitivo a sua utilização, foram usados padrões de concepção em várias partes da interface, uma que é comum a todas as páginas é a barra de navegação. Os outros padrões de concepção usados na aplicação estão especificados na respetivas secções abaixo. Este processo também foi inspirados em outras aplicações semelhantes, como *idealista*.
 - **Consistência** - na construção da interface fizemos o máximo para modularizar os componentes que a compõe, estes componentes são depois reutilizados onde necessário, criando uma aplicação com grande consistência interna.
- Flexibilidade:
 - **Dialogue initiative** - Em aspetos que consideramos ser mais ambíguos para o comum utilizador, providenciamos sucintas explicações ao utilizador.
 - **Customizability** - Como a aplicação foi desenvolvida tendo em mente, que seria para plataformas *mobile* e *desktop*, especialmente para o utilizador, foi implementado um *Responsive Web Design* para garantir a correta formatação das páginas para o tamanho do ecrã ou janela.
- Robustez:
 - **Task conformance** - O sistema foi desenhado para simplificar ao máximo as tarefas do utilizador. Isto foi alcançado usando padrões de concepção como *checkboxes* ou calendário para escolher datas, etc.
 - **Observability** - Nos testes da aplicação confirmamos que esta é fácil de navegar podendo chegar a todos os estados possíveis no máximo de dois cliques e o mesmo estado é simples de navegar para realizar as tarefas

9.1 Condutor

As tarefas deste utilizador são editar e visualizar a sua informação, realizar reservas e utilizar os *QRCodes*.

9.1.1 Barra de Navegação

Neste componente, comum a todas as páginas do utilizador condutor, é onde o mesmo volta a página principal, clicando no logótipo, acede ao seu perfil, abre o *QRCode* associado à sua conta, com o propósito de ocupar parques instantâneos, pode ver as suas notificações e sair da sua sessão.



Figura 9.1: Barra de Navegação

9.1.2 Página Principal

Na página principal do condutor, o utilizador poderá ver a listagem de parques com várias informações relevantes e realizar a tarefa de agendar uma reserva. Para encontrar o parque que lhe agrade, a interface contem filtros que o permitirão especificar características do parque e ordenar os resultados pelo preço e distância.

A screenshot of the ParkFinder main page. It displays three parking options: 'PARQUE VISCONDE DO RAIO' (797 m), 'B&B BRAGA LAMAÇÃES' (2.7 km), and 'BRAGA PARQUE' (1.1km). Each listing includes a thumbnail image, distance, estimated cost, and a 'Reservar' button. To the right, a sidebar titled 'Filtros' allows users to set filters for start date, end date, park type (Covered or Exterior), services (Electric vehicle charging, Multiple entrances/exits, Fast access, Lift), and reservations (Available reservation). A 'Aplicar Filtros' button is at the bottom.

Figura 9.2: Página Principal

Task conformance - procurar parque

QRCode

Ao clicar no icon de *QRCode* da página principal, o seguinte *popup* deve aparecer.

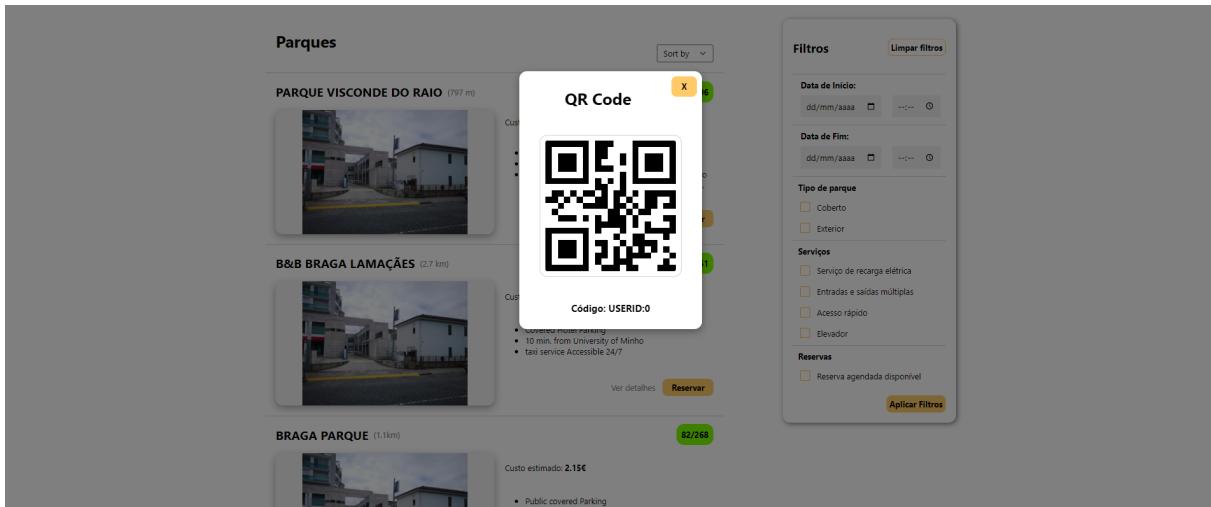


Figura 9.3: Código QR gerado com o ID do condutor.

Reserva

Ao clicar no botão de reservar um parque, o condutor depará-se com um *popup* que o permitirá a reserva, podendo esta ser agendada ou instantânea.

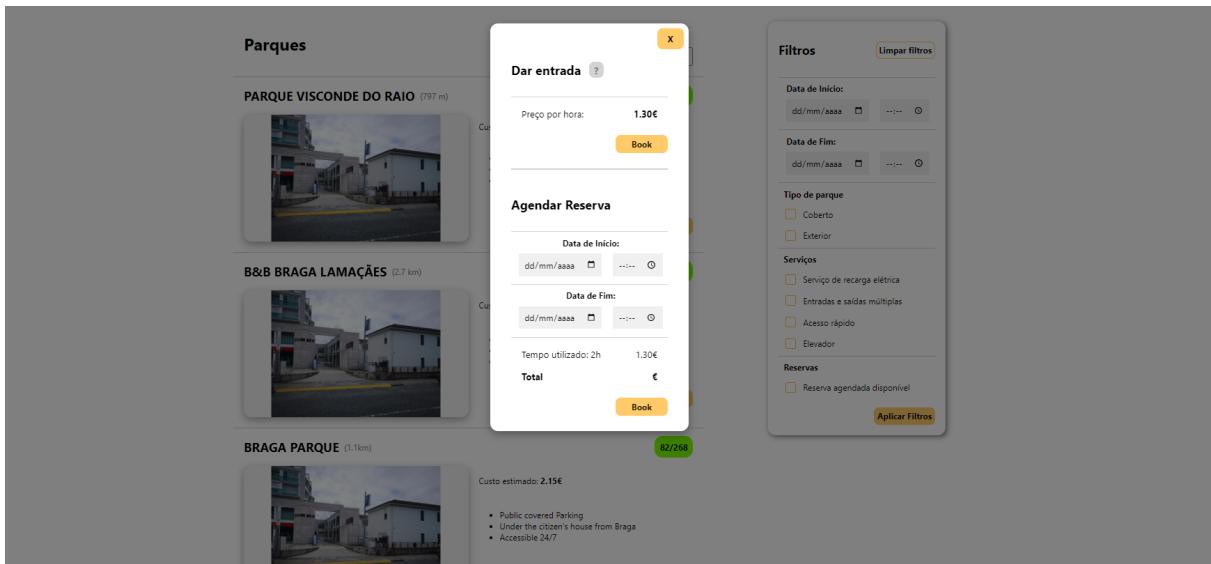


Figura 9.4: Aparência de uma reserva agendada.

Task conformance

9.1.3 Perfil

No perfil, o condutor poderá ver várias informações relacionadas consigo mesmo, e editar algumas delas.



Figura 9.5: Perfil do Condutor

Edição de Informações Pessoais

Ao clicar no botão de Editar Perfil, o condutor irá deparar-se com os campos que pode mudar da informação pessoal.

This screenshot shows the 'Editar Perfil' (Edit Profile) page. It includes a placeholder for a profile photo and a color selection tool for the background. The page lists personal details:

- Nome:** Pedro Jorge
- Email:** pedrojorge@gmail.com
- Número de Telefone:** 936978575

Below these fields are several buttons:

- Editar Perfil** (highlighted with a red border)
- Segurança**
- Minhas Reservas**
- Historico de Reservas**
- Historico de Pagamentos**

The right side of the screen contains a form with input fields for Name, Email, NIF, and Phone Number, along with a 'Gravar' (Save) button.

Figura 9.6: Edição da informação pessoal

Task conformance

Segurança

Ao clicar no botão de Segurança, o condutor poderá mudar a sua palavra-passe.

Figura 9.7: Mudança da palavra-passe.

Reservas Atuais

Nas Minhas Reservas, o utilizador verá a lista de reservas agendadas ainda por acontecer.

Figura 9.8: Reservas atuais do condutor

Também é onde o QRCode associado à reserva agendada pode ser utilizado ao clicar no botão Utilizar, fazendo aparecer um *popup* do mesmo.

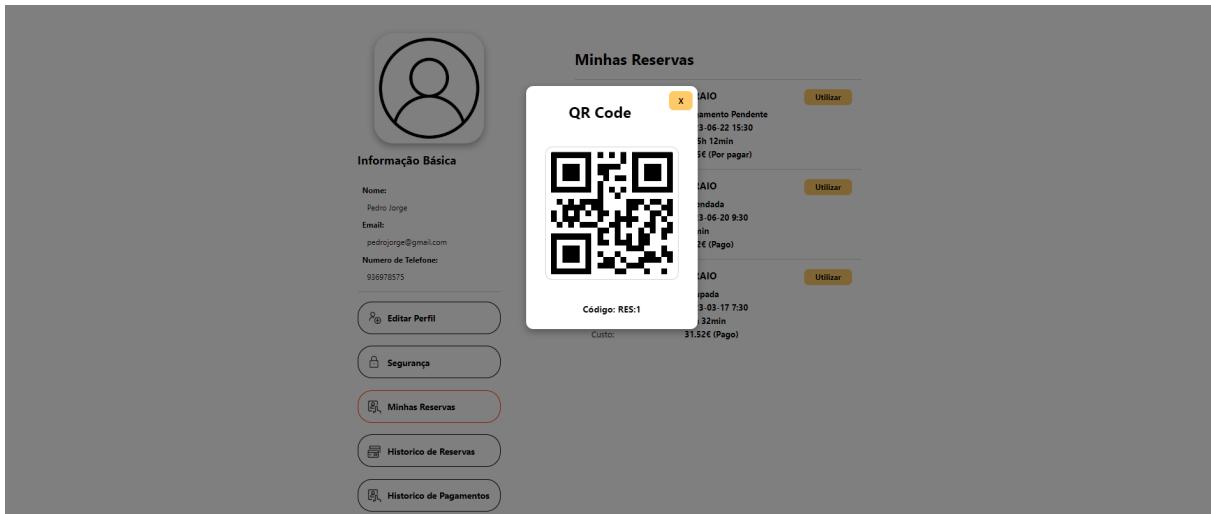


Figura 9.9: Código QR gerado com o ID da reserva.

Task conformance

Histórico de Pagamentos

Na opção Histórico de Pagamentos, o condutor poderá ver todas as transações que fez através da aplicação.

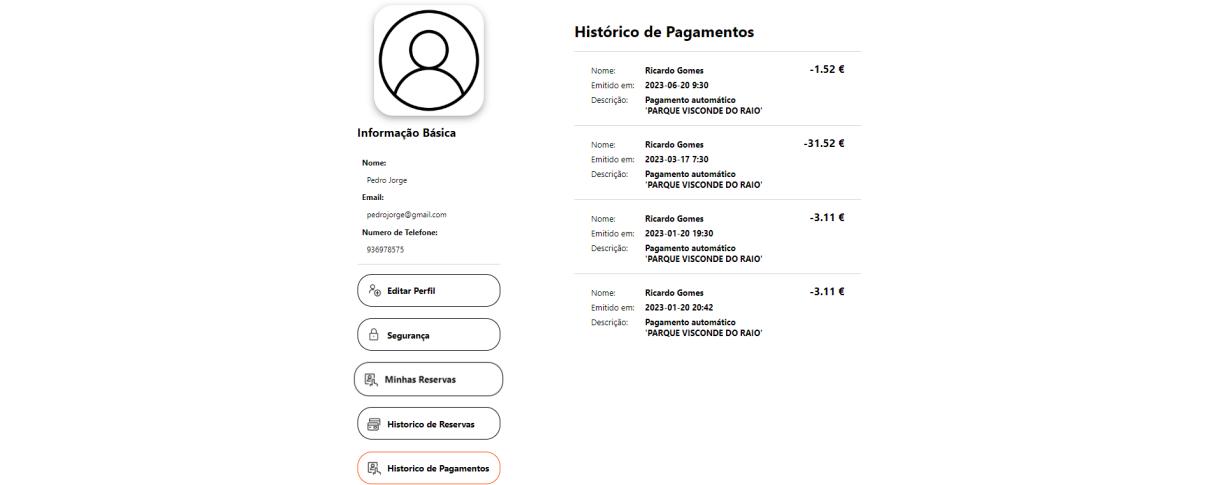


Figura 9.10: Histórico dos pagamentos efetuados.

Histórico de Reservas

Na opção Histórico de Reservas, o condutor poderá ver todas as reservas que fez na aplicação.

The screenshot shows a user profile with a placeholder icon and basic information: Name (Pedro Jorge), Email (pedrojorge@gmail.com), and Phone Number (936978575). Below the profile are five navigation buttons: 'Editar Perfil' (Edit Profile), 'Segurança' (Security), 'Minhas Reservas' (My Reservations), 'Histórico de Reservas' (Reservation History) which is highlighted in orange, and 'Histórico de Pagamentos' (Payment History).

Histórico de Reservas

PARQUE VISCONDE DO RAIO

Estado da Reserva:	Pagamento Pendente
Hora de entrada:	2023-03-22 15:30
Duração:	26:51 12min
Matrícula do veículo:	XA-21-AZ
Custo:	4.35€ (Por pagar)

PARQUE VISCONDE DO RAIO

Estado da Reserva:	Agendada
Hora de entrada:	2023-06-20 9:30
Duração:	12min
Matrícula do veículo:	XA-21-AZ
Custo:	1.52€ (Pago)

PARQUE VISCONDE DO RAIO

Estado da Reserva:	Ocupada
Hora de entrada:	2023-03-17 7:30
Duração:	12h 32min
Matrícula do veículo:	XA-21-AZ
Custo:	31.52€ (Pago)

PARQUE VISCONDE DO RAIO

Estado da Reserva:	Ocupada
Hora de entrada:	2023-03-17 7:30
Duração:	12h 32min
Matrícula do veículo:	XA-21-AZ
Custo:	31.52€ (Pago)

PARQUE VISCONDE DO RAIO

Figura 9.11: Histórico das reservas efetuadas.

9.1.4 Página De Parque Detalhada

Esta página é alcançada através do botão Ver detalhes de um parque da página principal, permitindo ver em melhor resolução a sua foto e outros detalhes do mesmo, sendo estes a descrição, onde pode ser visto a descrição atribuída a este pelo gestor e a distância ao utilizador, e as suas características, como o horário de funcionamento e o preçoário.

Descrição

PARQUE VISCONDE DO RAIO

Descrição

Informação

- Public covered parking
- 7 min. walk from the heart of the city
- Accessible from Monday to Friday from 8:00 am to 8:00 pm and Saturdays from 10:00 am to 8:00 pm.

Morada

- rua dos reis (797 m) de si

Agendar Reserva

Data de Início: dd/mm/aaaa

Data de Fim: dd/mm/aaaa

Tempo utilizado: 2h 1.30€

Total €

Book

Dar entrada

Preço por hora: 1.30€

Book

Figura 9.12: Página de parque detalhada com descrição.

Características

The screenshot shows a parking facility named 'PARQUE VISCONDE DO RAIO' located 'rua dos reis'. It features a small image of the parking area. On the right, there's a 'Book' button and a 'Agendar Reserva' (Schedule Reservation) form with fields for 'Data de Início' (Start Date) and 'Data de Fim' (End Date), both set to 'dd/mm/aaaa'. Below this, it shows 'Tempo utilizado: 2h' (Used time: 2h) and 'Total 1.30€'. A 'Book' button is also present here. To the left of the booking form, there's a 'Dar entrada' (Enter) section with a 'Preço por hora:' (Price per hour:) of '1.30€' and a 'Book' button. At the bottom left, there's a 'Características' (Characteristics) tab selected, showing operating hours and capacity information.

Horário de funcionamento

segunda:	07h30 - 13h00	14h30 - 21h00
terça:	07h30 - 13h00	14h30 - 21h00
quarta:	07h30 - 13h00	14h30 - 21h00
quinta:	07h30 - 13h00	14h30 - 21h00
sexta:	07h30 - 13h00	14h30 - 23h00
sábado:	08h00 - 13h00	
domingo:	Closed	

Capacidade

- Nº de lugares vagos: 45
- Nº de lugares total: 96

Figura 9.13: Página de parque detalhada com caraterísticas.

9.2 Administrador

O administrador tem como tarefas, gerir o número disponível de lugares, tendo atenção aos seus tipos, instantâneo e agendado, e o registo da entrada e saída de condutores que querem usufruir dos lugares instantâneos.

9.2.1 Cabeçalho

Este cabeçalho é comum a todos os utilizadores que são administrador, gestor e programador, e clicando no logótipo dirige o utilizador para a sua página principal e possui o botão de Sair da sessão.



Figura 9.14: Cabeçalho da página principal do Administrador

9.2.2 Página Principal

Nesta página, o administrador pode ver uma listagem dos parques que está encarregue com formato simples.

Parques

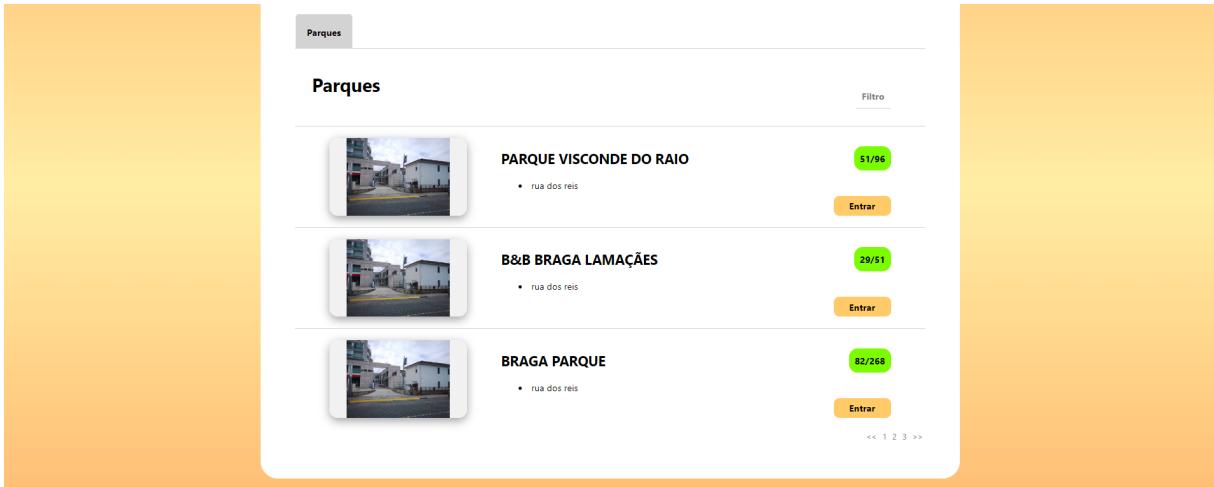


Figura 9.15: Corpo da página principal do Administrador

Outro aspecto que é comum aos administradores, gestores e programadores, é a possibilidade de filtrar os parques na forma de uma tabela de filtros colapsável.

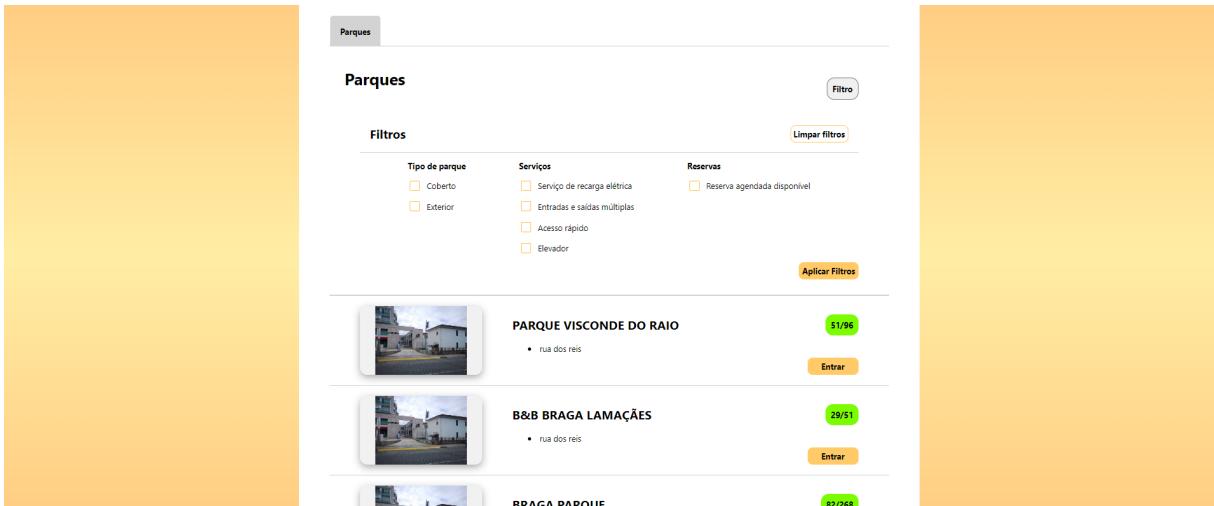


Figura 9.16: Opções de filtragem

9.2.3 Página De Parque Detalhada

Quando clica no botão Entrar de um parque é dirigido para esta página, onde pode fazer as suas funções.

Edição



Figura 9.17: Edição de lugares do Parque

Parque Instantâneo



Figura 9.18: Opção de parque instantâneo



Figura 9.19: Leitor de QRCode para admitir entrada no parque.

9.3 Gestor

O papel destes utilizadores é a visualização e estudo das estatísticas de cada parque, a criação e gestão de administradores, podendo alterar os encarregados pela administração dos parques. Também tem como função a gestão de certa informação que está associada a cada parque, sendo esta o estado de acessibilidade, ou seja, se está fechado ou aberto, o seu horário de disponibilidade, a imagem que refere o parque, o preçoário e a sua descrição.

9.3.1 Página Principal

Na página principal está presente umas abas de navegação, parques, administradores e estatísticas, as entidades que dizem respeito ao gestor, e é nela que o gestor navega, podendo realizar a tarefa de gerir os administradores e estudar as estatísticas.

Parques

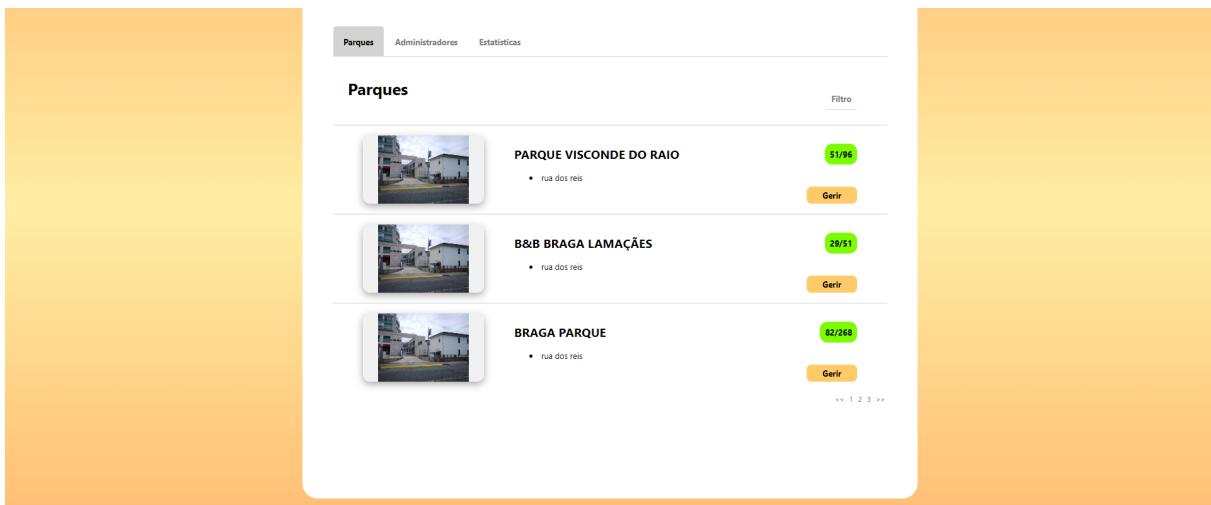


Figura 9.20: Página principal do gestor

Administradores

The screenshot shows a list of three administrators under the heading 'Administradores'. Each profile includes the administrator's name, email, mobile number, and the parks they manage. Buttons for managing these entries are also present.

Rui Antunes Gil Gomes de Sá Ribeiro Martins	Criar Administrador	
Email: joao@gmail.com Nº Telemóvel: 936978575	Parques: • BRAGA PARQUE • B&B BRAGA LAMAÇÃES	Remove Parque Adicionar Parque Remover
Rui Antunes Gil Gomes de Sá Ribeiro Martins	Criar Administrador	
Email: miguel@gmail.com Nº Telemóvel: 936978575	Parques: • BRAGA PARQUE • B&B BRAGA LAMAÇÃES	Remove Parque Adicionar Parque Remover
Rui Antunes Gil Gomes de Sá Ribeiro Martins	Criar Administrador	
Email: antonio@gmail.com Nº Telemóvel: 936978575	Parques: • BRAGA PARQUE • B&B BRAGA LAMAÇÃES	Remove Parque Adicionar Parque Remover

Figura 9.21: Gestão dos administradores

Estatísticas

The screenshot displays statistical data for three parks: PARQUE VISCONDE DO RAIO, B&B BRAGA LAMAÇÃES, and BRAGA PARQUE. Each park section lists three items, each preceded by a small icon.

PARQUE VISCONDE DO RAIO	B&B BRAGA LAMAÇÃES
• Stat: 5	• Stat: 5
• Stat: 5	• Stat: 5
• Stat: 5	• Stat: 5

BRAGA PARQUE
• Stat: 5
• Stat: 5
• Stat: 5

Figura 9.22: Estatísticas dos parques do Gestor

Adicionar Parques ao Administrador

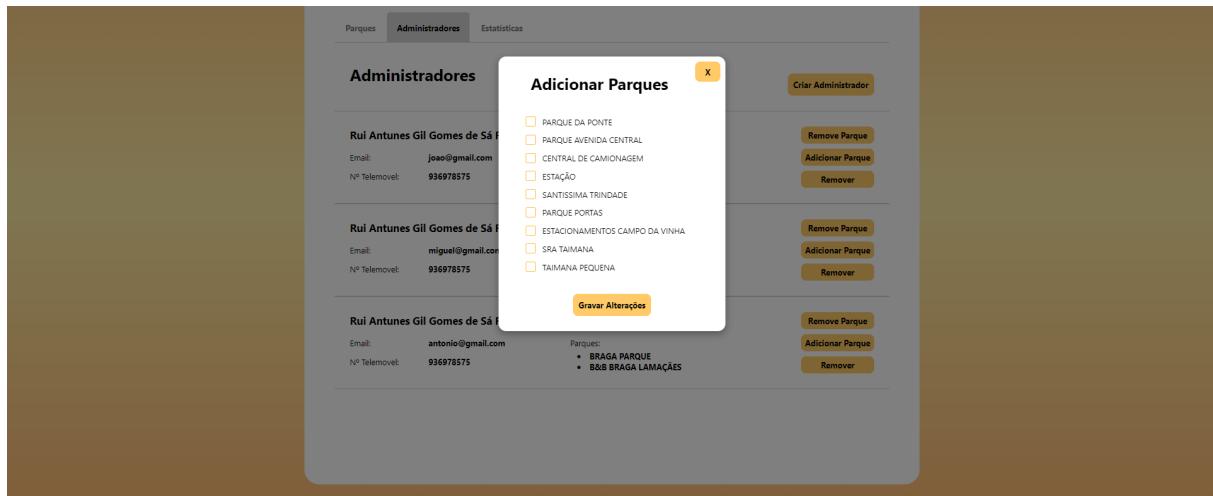


Figura 9.23: Adicionar parques ao administrador

9.3.2 Página De Parque Detalhada

Nesta página, que é acedida através do botão Gerir dos parques na página principal, e é nela que o gestor pode, outra vez, consultar as estísticas do certo parque e editar as informações do mesmo.

Estatística



Figura 9.24: Estatísticas do parque na página detalhada do mesmo.

Multithreading

Edição

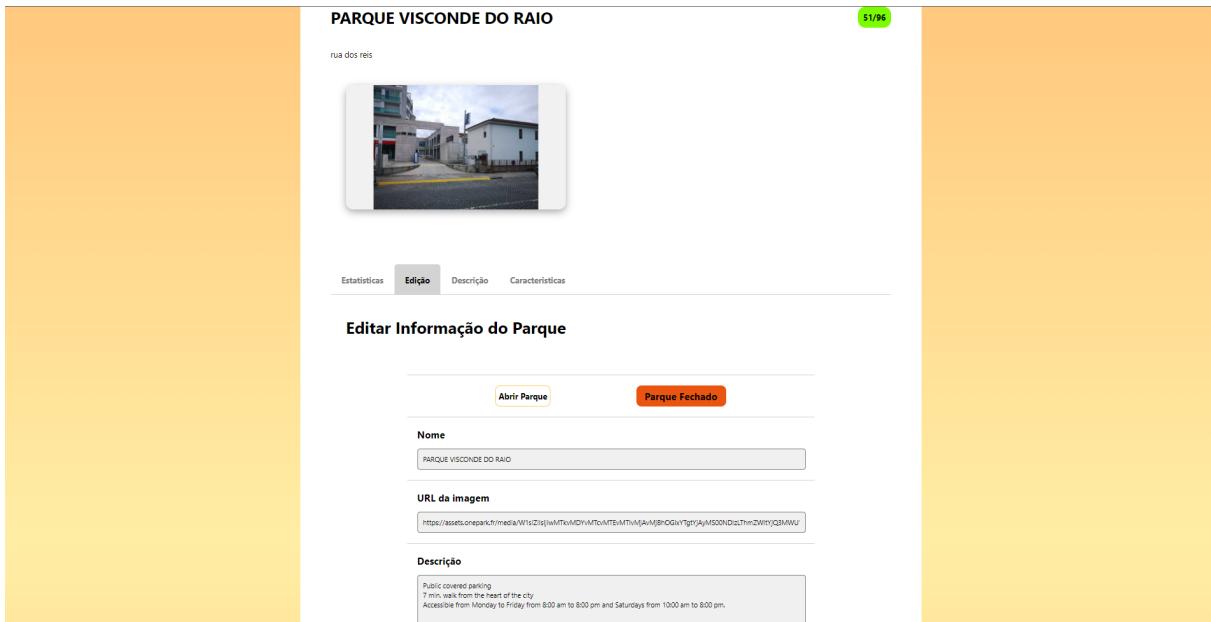


Figura 9.25: Edição da informação do Parque

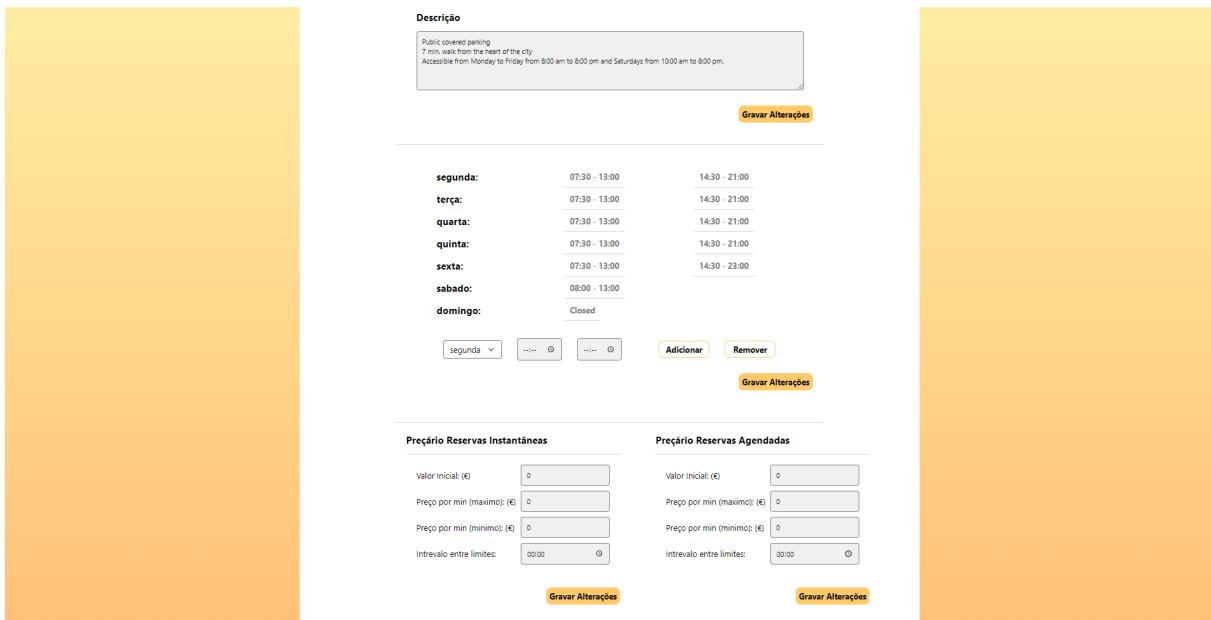


Figura 9.26: Edição da informação do parque

Task conformance

9.3.3 Página De Criação de Administrador

Parques Administradores Estatísticas

Criar Administrador

Nome

Email

NIF

Número de Telemóvel

Gravar

Figura 9.27: Criação de administrador

Multithreading

9.4 Programador

O programador tem como funções a criação e gestão das entidades parque e gestor, podendo alterar os encarregados pelas gestões dos parques, parecido a como os gestores conseguem alterar a administração do parque.

9.4.1 Página Principal

O programador tem uma página principal com duas abas, parques e gestores, que são as entidades que pode criar, podendo fazer as suas funções muito intuitivamente.

Parques

Parques Gestores

Parques

Criar Parque

Filtro

PARQUE VISCONDE DO RAIO
• rua dos reis 51/96 Remover

B&B BRAGA LAMAÇÃES
• rua dos reis 29/51 Remover

BRAGA PARQUE
• rua dos reis 82/268 Remover

<< 1 2 3 >>

Figura 9.28: Página principal programador

Criação de Parque

The screenshot shows a web-based application for creating a park. At the top, there are two tabs: 'Parques' (selected) and 'Gestores'. Below the tabs, the title 'Criar Parque' is displayed. The form consists of several input fields and buttons:

- Nome:** A text input field labeled 'Nome'.
- Morada:** A text input field labeled 'Email'.
- Localização do parque:** Two input fields for 'Latitude' (0) and 'Longitude' (0), followed by a 'Gravar' (Save) button.

Figura 9.29: Criação de parque

Gestores

The screenshot displays a list of managers with their details and associated parks. Each entry includes an 'Adicionar Parque' (Add Park) and 'Remover' (Remove) button.

Gestores			
Rui Antunes Gil Gomes de Sá Ribeiro Martins	rui@gmail.com 936978575	Parques: • BRAGA PARQUE • B&B BRAGA LAMAÇAS	Criar Parque Adicionar Parque Remover
Rui Antunes Gil Gomes de Sá Ribeiro Martins	carlos@gmail.com 936978575	Parques: • Alex_Nao • Sabe_Conduzir	Criar Parque Adicionar Parque Remover
Rui Antunes Gil Gomes de Sá Ribeiro Martins	pedro@gmail.com 936978575	Parques: • esquina	Criar Parque Adicionar Parque Remover

Figura 9.30: Informação dos gestores

Criação de Gestor

The screenshot shows a user interface for creating a manager. At the top, there are two tabs: 'Parques' and 'Gestores', with 'Gestores' being the active tab. Below the tabs is a section titled 'Criar Gestor'. This section contains four input fields: 'Nome' (Name), 'Email', 'NIF', and 'Número de Telemóvel' (Mobile Number). Each field has a placeholder text ('Nome', 'Email', '0', '0') and a corresponding text input box. To the right of the 'Número de Telemóvel' input is a small orange 'Gravar' (Save) button.

Figura 9.31: Criação de gestor

9.5 Comentário sobre as Shneiderman's 8 Golden Rules

1. *Strive for consistency* - Esta regra foi a melhor aplicada na nossa aplicação, havendo um grande grau de consistência, seja nas tarefas, seja na aparência da interface;
2. *Enable frequent users to use shortcuts* - Como todas as tarefas conseguem ser feitas no mínimo em três cliques do rato, onde não é necessário grande experiência, consideramos que o cumprimento desta regra teria pouco impacto;
3. *Offer informative feedback* - Esta regra foi cumprida ao utilizar balões de fala em aspectos da interação que podem ser mais ambíguos para o utilizador.
4. *Design dialogs to yield closure* - Esta regra foi cumprida no sentido que a maioria das ações podem ser confirmadas no momento seguinte na mudança da interface para as refletir, ou podem ser confirmadas em páginas rapidamente alcançáveis.
5. *Offer error prevention and simple error handling* - Cumprimos esta, limitando os *inputs* ao necessário, e avisando quando alguma ação não é válida.
6. *Permit easy reversal of actions* - Esta regra está a ser cumprida em vários atos de gestão de várias entidades, como a alteração da atribuição dos parques aos gestores ou administradores.
7. *Support internal locus of control* - Tendo que os utilizadores tem liberdade total na utilização da aplicação dentro do limite dos que lhe é permitido, consideramos esta regra cumprida.
8. *Reduce short-term memory load* - As informações necessárias para cada tarefa, estão sempre no campo de visão do utilizador, portanto consideramos que esta regra esteja cumprida

9.6 Planos para o futuro

No futuro, temos planos para implementar a recuperabilidade dos princípios de usabilidade, pois como há ações que têm grande impacto no estado dos dados, ao assegurar la, estariam os a aumentar a qualidade do produto final num elevado grau. Esta aplicação foi originalmente

concebida para ser suportado em plataformas *mobile* e *desktop*, usando *Responsive web design*, e embora o tivemos usado, reconhecemos que ainda tem grande potencial para melhorias

10. Avaliação da Solução

Quanto à unidade curricular de Arquiteturas Aplicacionais podemos afirmar que utilizamos para o desenvolvimento da solução alguns dos conceitos lecionados na unidade curricular. Como a abordagem de desenvolvimento de software Model Driven Development que nos premitiu a geração de código Java de forma automática a partir do diagrama PSM. Também utilizamos Beans, embora, tenhamos usado os Beans disponibilizados pela *framework* Spring que não são iguais em termos de funcionalidade aos Java Beans que foram lecionados nas aulas. Acreditamos que conseguimos demonstrar o conhecimento adquirido nesta matéria com o uso de Spring Beans. Por outro lado, também usamos Hibernate com anotações para persistir as classes com informação relevante na base de dados.

Para unidade curricular de Sistemas Interativos Confiáveis podemos afirmar que utilizamos para o desenvolvimento da solução alguns dos conceitos lecionados na unidade curricular. Como a abordagem de desenvolvimento de interface seguimos os princípios de usabilidade que nos premitiu a geração de código Java de forma automática a partir do diagrama PSM. Também utilizamos Beans, embora, tenhamos usado os Beans disponibilizados pela *framework* Spring que não são iguais em termos de funcionalidade aos Java Beans que foram lecionados nas aulas. Acreditamos que conseguimos demonstrar o conhecimento adquirido nesta matéria com o uso de Spring Beans. Por outro lado, também usamos Hibernate com anotações para persistir as classes com informação relevante na base de dados.

Em conclusão, o grupo considera que conseguiu utilizar os conhecimentos adquiridos em ambas as unidades curriculares de modo a obter a solução ideal. Decidimos desenvolver a camada de apresentação separadamente da camada da lógica de negócios e dados, mas ao mesmo tempo de forma simultânea. Deste modo, fomos capazes de atingir um meio termo onde consideramos os benefícios e malefícios de certa mudança de design em ambas as partes do projeto. Dito isto, tivemos algumas dificuldades a integrar as duas partes do projeto e, por isso, mesmo tendo ambas as partes bastante desenvolvidas só conseguimos integrar as camadas para algumas funcionalidades principais até a data da apresentação.