



Ano Letivo 2023/2024 | 2º Semestre

CodeWave

Aplicação web para um parque aquático

Projeto Temático de Aplicações Web

Licenciatura em Tecnologias da Informação

Autores:

Artiom Masliucov | 109295

Bruno Migueis | 102238

Daniel Silva | 113900

Gabriel Cravo | 87689

Lucas Duarte | 109190

Miguel Pirré | 114017

Grupo 1

Águeda | 10 de junho de 2024





Ano Letivo 2023/2024 | 2º Semestre

CodeWave

Aplicação web para um parque aquático

Projeto Temático de Aplicações Web

Licenciatura em Tecnologias da Informação

Autores:

Artiom Masliucov | 109295 | arti@ua.pt

Bruno Migueis | 102238 | brunomigueis@ua.pt

Daniel Silva | 113900 | d.m.silva@ua.pt

Gabriel Cravo | 87689 | gcravo@ua.pt

Lucas Duarte | 109190 | lucasduarte2@ua.pt

Miguel Pirré | 114017 | pirre@ua.pt

Professor Orientador:

Rui Isidro

Grupo 1

Águeda | 10 de junho de 2024

Agradecimentos

Este projeto representa a concretização de um percurso desafiante e é fruto do nosso empenho e da colaboração de todos os membros do grupo. Contudo, a sua realização não teria sido possível sem o valioso contributo e apoio de algumas pessoas que, direta ou indiretamente, se envolveram neste projeto.

Expressamos a nossa profunda gratidão ao Professor Rui Orlando Gomes Isidro, nosso orientador, cuja sabedoria e disponibilidade constante para nos acompanhar passo a passo foram fundamentais para o sucesso deste trabalho. A sua experiência e excelência na supervisão foram elementos-chave que nos guiaram ao longo desta jornada académica.

Estendemos também os nossos sinceros agradecimentos a toda a equipa docente do nosso curso, cujo suporte inestimável e ensinamentos foram essenciais não só para a realização deste trabalho, mas também para o desenvolvimento da nossa metodologia de trabalho e crescimento intelectual.

A todos estes, à ESTGA, e a todas as pessoas que, de alguma forma, marcaram a nossa trajetória e contribuíram para o nosso crescimento profissional e pessoal, o nosso mais sincero obrigado.

Um grande bem-haja a todos que foram parte desta importante etapa da nossa vida académica.

Resumo

O presente relatório documenta meteticulosamente o processo de desenvolvimento de uma aplicação web para um parque aquático, “CodeWave”, seguindo a metodologia em cascata. Realizado no âmbito da Licenciatura em Tecnologias da Informação da Escola Superior de Tecnologia e Gestão de Águeda, este projeto foi orientado pelo docente Rui Isidro durante o segundo semestre do segundo ano.

O projeto iniciou-se com uma fase de planeamento rigorosa, na qual se estabeleceu o objetivo, com base no tema escolhido, e se analisaram os requisitos do sistema para a construção do modelo conceptual. Identificaram-se as funcionalidades, os casos de utilização, os utilizadores, bem como os requisitos funcionais, não funcionais e de hardware.

Na fase de desenho, desenvolveu-se o diagrama de casos de utilização, o esquema relacional da base de dados e o protótipo de baixa fidelidade. Para o design do sistema, recorreu-se a tecnologias web como HTML, CSS e JavaScript para o *frontend* e PHP para o *backend*, de forma a garantir uma integração eficaz com a base de dados.

Seguiu-se a implementação, onde se construiu a base de dados, utilizando PostgreSQL, e desenvolveu-se o website, avançando-se para a fase de testes, para assegurar a funcionalidade e usabilidade da aplicação web. A documentação detalhada de cada etapa foi uma constante, culminando na redação do relatório final.

Este relatório finaliza com uma reflexão sobre o percurso do projeto, destacando as estratégias adotadas e propondo melhorias futuras para a aplicação, com o objetivo de otimizar a experiência do utilizador e a operacionalidade do parque aquático.

Índice geral

1.	Introdução	12
1.1	Enquadramento	12
1.2	Objetivos do Projeto	12
1.3	Público-alvo	13
2.	Estado da arte	14
2.1	Aplicações Relevantes	14
3.	Planeamento e Execução	19
3.1	Plano do projeto	19
3.2	Metodologia adotada	19
3.3	Calendarização das tarefas prevista	20
3.4	Calendarização das tarefas executada	21
3.5	Divisão de tarefas	23
4.	Modelo de requisitos	24
4.1	Requisitos funcionais	24
4.2	Restrições e requisitos não funcionais	27
4.3	Requisitos de interface e facilidade de uso	29
4.4	Requisitos de desempenho	29
4.5	Requisitos de segurança e integridade dos dados	30
4.6	Requisitos de interface com sistemas externos e ambientes de execução	30
5.	Normas e regulamentação específicas aplicáveis	31
5.1	Requisitos de hardware	33
6.	Diagrama de Arquitetura do Sistema	34
7.	Diagrama de atividades	35
7.1	Login para o gerente e funcionários	35
7.2	Login para utilizadores	36
8.	Casos de Utilização	38
8.1	Atores	38

8.2	Casos de utilização	39
8.3	Cobertura de requisitos.....	61
9.	Prototipagem baixa-fidelidade	63
10.	Análise de tecnologias	69
10.1	Bibliotecas Externas.....	75
11.	Desenvolvimento do website	77
11.1	Diagrama de Fluxo de Navegação do Website	77
11.1.1	Diagrama de Fluxo de Navegação do Website (Gerente)	77
11.1.2	Diagrama de Fluxo de Navegação do Website (Site).....	80
11.1.3	Diagrama de Fluxo de Navegação do Website (Loja-Online)	82
11.2	Modelo de Dados Persistente	84
12.	Interface	86
12.1	Página principal	86
12.2	Página do mapa interativo	88
12.3	Página contactos	89
12.4	Página horários	89
12.5	Página bilhetes.....	90
12.6	Página restauração	91
12.7	Página alojamento	92
12.8	Página alugueres.....	94
12.9	Página eventos	95
12.10	Página notícias.....	97
12.11	Página loja online	98
12.12	Página loja online produtos	99
12.13	Página <i>login e signup</i>	100
12.14	Página carrinho de compras.....	101
12.15	Página feedbacks	105
12.16	Dashboard admin.....	107
12.17	Dashboard – Bilhetes.....	108
12.18	Dashboard – Info Bilhetes	109
12.19	Dashboard – Alugueres.....	110

12.20	Dashboard – Categorias.....	111
12.21	Dashboard – Eventos	111
12.22	Dashboard – Zonas	112
12.23	Dashboard – Produtos.....	113
12.24	Dashboard – Funcionários	114
12.25	Dashboard – Cargos.....	114
12.26	Dashboard – Feedbacks	115
12.27	Dashboard – Restauração	116
12.28	Dashboard – Horários.....	117
12.29	Dashboard – Notícias.....	117
12.30	Dashboard – Progressão	118
12.31	Dashboard – Mapa.....	119
13.	Mapa de AR (Realidade Aumentada).....	120
14.	Controlador do Parque de Estacionamento	121
15.	Controlador de iluminação Arduino.....	122
16.	Testes e validação.....	126
16.1	Testes de sistema e de validação.....	126
16.2	Testes de usabilidade	127
17.	Análise de Resultados	132
18.	Reflexão Crítica e Conclusões	135
18.1	Atividades Desenvolvidas	135
18.2	Estratégias de Trabalho Adotadas.....	135
18.3	Planeamento Previsto e Cronograma Executado	136
18.4	Sugestões para o Futuro.....	136
18.5	Síntese das Experiências.....	137
18.6	Conclusão	138
19.	Bibliografia	139
ANEXO A	140
ANEXO B	146
ANEXO D	149
ANEXO E	155

Índice de tabelas

Tabela 1 - Requisitos funcionais	26
Tabela 2 - Restrições	27
Tabela 3 - Requisitos não funcionais	28
Tabela 4 - Requisitos de interface e facilidade de uso	29
Tabela 5 - Requisitos de desempenho	29
Tabela 6 - Requisitos de segurança e integridade dos dados.....	30
Tabela 7 - Requisitos de interface com sistemas externos e ambientes de execução.....	30
Tabela 8 - Requisitos de hardware computador	33
Tabela 9 - Requisitos de hardware servidor	33
Tabela 10 - Atores do sistema	38
Tabela 11 Caso de utilização gerir lugares de estacionamento	41
Tabela 12 Caso de utilização gerir secção de feedback	41
Tabela 13 Caso de utilização gerir reservas	42
Tabela 14 Caso de utilização gerir equipamentos	42
Tabela 15 Caso de utilização gerir produtos	43
Tabela 16 Caso de utilização gerir contas	43
Tabela 17 Caso de utilização gerir alertas.....	44
Tabela 18 Caso de utilização gerir alugueres	44
Tabela 19 Caso de utilização validar bilhete.....	45
Tabela 20 Caso de utilização autenticação.....	45
Tabela 21 Caso de utilização criar conta.....	46
Tabela 22 Caso de utilização verificar meteorologia	46
Tabela 23 Caso de utilização interagir com o mapa.....	47
Tabela 24 Caso de utilização consultar horários	47
Tabela 25 Caso de utilização agendar evento	48
Tabela 26 Caso de utilização alugar conveniências	48
Tabela 27 Caso de utilização comprar produto	49
Tabela 28 Caso de utilização comprar bilhete.....	49
Tabela 29 Caso de utilização verificar afluência do parque.....	50

Tabela 30 Caso de utilização consultar histórico de compras	50
Tabela 31 Caso de utilização gerir restauração	51
Tabela 32 Caso de utilização editar carrinho de compras	51
Tabela 33 Caso de utilização procurar produto	52
Tabela 34 Caso de utilização enviar feedback	52
Tabela 35 Caso de utilização gerir progressão de carreiras	53
Tabela 36 Caso de utilização visualizar menus da restauração	53
Tabela 37 Caso de utilização gerir horários	54
Tabela 38 Caso de utilização visualizar galeria	54
Tabela 39 Caso de utilização contactar o parque	55
Tabela 40 Caso de utilização alugar alojamento	55
Tabela 41 Caso de utilização ver notícias	56
Tabela 42 Caso de utilização gerir notícias.....	56
Tabela 43 Caso de utilização subscrever newsletter	57
Tabela 44 Caso de utilização gerir zonas	57
Tabela 45 Caso de utilização gerir categorias	58
Tabela 46 Caso de utilização gerir funcionários	58
Tabela 47 Caso de utilização gerir cargos.....	58
Tabela 48 Caso de utilização emitir relatório de estatísticas.....	59
Tabela 49 Caso de utilização visualizar mapa em RA	59
Tabela 50 Caso de utilização chat com integração de IA.....	60
Tabela 51 - Cobertura de requisitos	62
Tabela 52 Bibliotecas Externas	75
Tabela 53 Testes de usabilidade funcionário.....	128
Tabela 54 - Testes de usabilidade gerente	129
Tabela 55 - Requisitos cumpridos.....	132
Tabela 56 - Requisitos não cumpridos	134
Tabela 57 - Requisitos parcialmente cumpridos	134

Índice de figuras

Figura 1 - NorPark	15
Figura 2 - NaturWaterpark	15
Figura 3 - Parque Aquático de Amarante	16
Figura 4 - VagaSplash	17
Figura 5 - ZooMarine.....	18
Figura 6 - Calendarização Prevista	20
Figura 7 - Calendarização executada	21
Figura 8 - Divisão das tarefas	23
Figura 9 - Diagrama de Arquitetura do Sistema	34
Figura 10 - Diagrama de atividades de login para o gerente e funcionários.....	35
Figura 11 - Diagrama de atividades de login para os utilizadores	37
Figura 12 - Casos de utilização.....	39
Figura 13 - Diagrama de casos de utilização da loja online.....	40
Figura 14 - Homepage	64
Figura 15 - Horários.....	65
Figura 16 - Loja online	65
Figura 17 - Galeria.....	66
Figura 18 - Alugueres.....	66
Figura 19 - Mapa.....	67
Figura 20 - Dashboard gerente.....	68
Figura 21 - PgAdmin 4	69
Figura 22 - GitHub do projeto CodeWave	70
Figura 23 - Visual Studio Code.....	71
Figura 24 - Discord	72
Figura 25 - Teams	73
Figura 26 - Trello	73
Figura 27 - Balsamiq.....	74
Figura 28 - Figma.....	75
Figura 29 - Diagrama de Fluxo de Navegação do Website (Admin).....	79

Figura 30 - Diagrama de Fluxo de Navegação do Website (Site)	81
Figura 31 - Diagrama de Fluxo de Navegação do Website (Loja-Online).....	83
Figura 32 - Modelo de base de dados conceptual	84
Figura 33 - Modelo de base de dados relacional.....	85
Figura 34 - Página principal.....	87
Figura 35 - Página do mapa interativo	88
Figura 36 - Página contactos.....	89
Figura 37 - Página horários.....	90
Figura 38 - Página bilhetes	91
Figura 39 - Página restauração.....	92
Figura 40 - Página alojamento	93
Figura 41 - Página alugueres.....	94
Figura 42 - Página eventos.....	95
Figura 43 - Página criar evento	96
Figura 44 - Página eventos.....	97
Figura 45 - Página loja online	98
Figura 46 - Página loja online produtos	99
Figura 47 - Página login utilizador	100
Figura 48 - Página registo utilizador.....	100
Figura 49 - Página carrinho de compras	101
Figura 50 - Finalização da compra.....	102
Figura 51 - Bilhete eletrónico	103
Figura 52 - Fatura.....	104
Figura 53 - Página feedbacks	105
Figura 54 - Página com todos feedbacks	106
Figura 55 - Dashboard admin.....	107
Figura 56 - Dashboard bilhetes	108
Figura 57 - Dashboard info bilhetes.....	109
Figura 58 - Dashboard alugueres	110
Figura 59 - Dashboard categorias	111
Figura 60 - Dashboard eventos	112

Figura 61 - Dashboard zonas	113
Figura 62 - Dashboard produtos.....	113
Figura 63 - Dashboard funcionários.....	114
Figura 64 - Dashboard cargos	115
Figura 65 - Dashboard feedbacks.....	115
Figura 66 - Dashboard restauração	116
Figura 67 - Dashboard horários	117
Figura 68 - Dashboard notícias	117
Figura 69- Dashboard progressão	118
Figura 70 - Dashboard zonas	119
Figura 71 - Mapa Realidade Aumentada.....	120
Figura 72 - Diagrama de estacionamento do parque (Arduino).....	121
Figura 73 - Diagrama de iluminação (Arduino).....	122
Figura 74 - Código controlador de leds	123
Figura 75 - Protótipo do controlador de iluminação	124
Figura 76 - Protótipo do controlador de iluminação	125
Figura 77 - Documento de consentimento de participação no teste.....	130
Figura 78 - Questionário de satisfação.....	131
Figura 79 - Gráfico dos requisitos cumpridos, não cumpridos e parcialmente cumpridos	132

Lista de siglas e abreviaturas

ESTGA	Escola Superior de Tecnologia e Gestão de Águeda
RF	Requisitos funcionais
Ref	Referência
RSeg	Requisito de segurança e integridade
RDes	Requisito de desempenho
RInt	Requisito de interfaces e facilidades de uso
RNF	Requisito não funcional
RI	Requisito de interface com sistemas externos e ambientes de execução
PBF	Protótipo de baixa-fidelidade
RFID	Radio Frequency Identification
IDE	Integrated Development Environment
Código QR	Código <i>Quick Response</i>
UI	Interface de Utilizador
UX	Experiência do Utilizador
RA	Realidade Aumentada
Sensor PIR	Sensor infravermelho passivo

1. Introdução

No âmbito do Módulo Temático em Aplicações Web, do segundo semestre do segundo ano da Licenciatura em Tecnologias da Informação, da Escola Superior de Tecnologias e Gestão de Águeda, o qual engloba, para além do Projeto Temático em Aplicações Web, a unidade curricular associada: Programação e Tecnologias Web, foi proposto elaborar um website, utilizando *frontend* e *backend* (*fullstack*), que tivesse conexão e manipulação com uma base de dados. Decidimos então, em grupo, dar origem ao **CodeWave**, um website de um estabelecimento na área das diversões aquáticas que também permite toda a sua gestão. Neste relatório serão debatidos e abordados os seguintes capítulos para além da própria Introdução: Planeamento e Execução do trabalho; Levantamento do Estado de Arte; Esquema Funcional e modelo de Requisitos; Casos de Utilização; Análise de Tecnologias; Modelo de dados Persistente e Modelo estrutural; Desenvolvimento da aplicação; Análise de Resultados; Reflexão Crítica e Conclusão.

É ainda composto por uma Bibliografia e Anexos. Estes pretendem complementar a informação apresentada nos capítulos anteriores.

1.1 Enquadramento

Existe vários estabelecimentos de grande renome em Portugal na área das diversões aquáticas, porém após visitar os seus websites percebemos que cada um tem um forte, mas não estão completos com informações importantes e essenciais. O nosso objetivo com o **CodeWave** é criar uma página web com uma excelente experiência visual, uma organização otimizada e que agilize o processo tanto do lado do cliente como do lado do staff. Reduzindo os erros, aumentando a eficiência e a satisfação do cliente. Um sistema como o nosso, simples e prático de utilizar no dia a dia, pode ser bastante útil para responder a esses problemas.

1.2 Objetivos do Projeto

O sistema resultante permitirá uma grande melhoria face à não utilização deste, nos pontos referentes a:

Eficiência – A aplicação permitirá aos funcionários agilizar o processo dos bilhetes, das reservas, da venda do *merchandising* e dos alojamentos. Estes poderão alterar todas as informações relativamente ao parque de forma rápida e precisa, devido à sua organização, conseguindo assim uma manutenção mais fácil e eficiente, reduzindo o risco de erros. O website conta também com uma interface do lado do cliente que é eficiente, permite que se navegue e encontre todas as informações necessárias para escolher o melhor dia para visitar o nosso parque.

Gestão de Afluência – O website possibilitará gerir a afluência do parque, com essa informação o gerente sabe quantos bilhetes mais podem ser vendidos. O próprio website limita também a venda de bilhetes em dias de eventos permitindo assim que o parque mantenha um ótimo funcionamento mesmo em dias de maior afluxo. Isto proporcionará uma melhor gestão de staff, monitorização e capacidade do estabelecimento.

Pagamentos – O website permitirá realizar compras de *merchandising* e alugueres tanto online como pessoalmente, criando estas possibilidades, é tirado stress do staff do parque e consequentemente, melhorando o tempo das filas. A entrega do *merchandising* também poderá ser entregue em casa através da nossa transportadora tornando assim possível com que não se tenha de deslocar ao local do parque para receber o *merchandising* comprado online.

Estatísticas – A aplicação possibilitará, ao gerente, ter acesso a diversas estatísticas relevantes, tais como gráficos de faturação por mês. O que não só facilitará a gestão do estabelecimento, como também levará a uma visão mais realista do que os clientes pretendem e gostam.

1.3 Público-alvo

O **CodeWave** é projetado para atender a todos os tipos de visitantes, desde famílias com crianças até grupos de amigos e turistas. O nosso objetivo é oferecer uma variedade de atrações e serviços que satisfaçam as necessidades e expectativas de cada visitante, proporcionando diversão, segurança e conforto. Com uma interface de reservas intuitiva e uma gestão eficiente, garantimos que todos os nossos clientes desfrutem de uma experiência inesquecível. Queremos criar momentos memoráveis para cada pessoa que nos visita, enquanto aumentamos a eficiência operacional do nosso parque.

2. Estado da arte

Em diversas partes do mundo, em que têm surgido websites inovadores que implementam novos sistemas e tecnologias com o objetivo de auxiliar e automatizar a gestão e o monitoramento de parques aquáticos. Estes avanços tecnológicos não só facilitam o trabalho dos gestores, como também melhoram a experiência dos visitantes.

2.1 Aplicações Relevantes

Numa fase inicial do projeto teve-se como objetivo analisar as soluções já existentes no mercado e estudar como são implementadas. Foram extraídas ideias e funcionalidades úteis dessas soluções para melhorar o desenvolvimento do trabalho.

Identificaram-se websites que desempenham a mesma função que o nosso, considerou-se importante a sua identificação. Esses websites mostraram-nos as funcionalidades que ao implementarmos melhorava toda a dinâmica do nosso projeto. O NorPark mostrou-nos a necessidade de manter um design limpo e fácil de usar, assim como a necessidade de divulgação das nossas redes sociais para um maior alcance de público. O NaturWaterPark tornou-se um ponto de inspiração nossa no que toca à criação do nosso próprio *merchandising* e ao seu sistema de reservas aparentemente simples e eficiente. O website do Parque Aquático de Amarante destaca-se pelo seu design atraente, a sua página de notícias e o vasto leque de funcionalidades. O VagaSplash revelou-se importante pois dá ao cliente todas as informações sobre os serviços realizados no parque e pela possibilidade de organização de festas e eventos dentro do parque. Por último o Slide & Splash, têm implementado um mapa 3D de todo o parque permitindo assim explorar todas as diversões no conforto da casa do cliente, assim como a educação ambiental que está sempre presente pelo website.



FIGURA 1 - NORPARK

NorPark

A análise do site do NorPark oferece diversas oportunidades de melhoria que podem ser incorporadas no desenvolvimento do projeto CodeWave. A estrutura de navegação e a tabela de preços podem servir de inspiração para criar uma interface organizada e informativa. Além disso, é possível explorar recursos interativos e criativos, como visualizações em 3D, e implementar uma loja de *merchandising* online para aumentar o envolvimento e a experiência dos visitantes.

Esta solução foi escolhida por possuir características que o grupo considerou fundamentais para o desenvolvimento do projeto. Uma característica desejada para integrar na solução proposta pelo grupo é a possibilidade de gestão de reservas online, permitindo assim uma organização eficiente dos recursos do parque e proporcionando uma melhor experiência para os visitantes.



FIGURA 2 - NATURWATERPARK

NaturWaterpark

A análise do site do NaturWaterpark revela várias oportunidades que podem ser aproveitadas no desenvolvimento do projeto CodeWave. A integração de um sistema de reservas online é uma funcionalidade crucial para a gestão eficaz das visitas. Programas para empresas, como atividades de *team building*, representam uma excelente oportunidade para diversificar e ampliar os serviços oferecidos. Adicionalmente, a oferta de alojamento e a inclusão de uma loja de *merchandising* online são elementos que podem aumentar o envolvimento dos visitantes e melhorar a sua experiência. Esta solução foi escolhida por conter características que o grupo considerou fundamentais para o desenvolvimento do projeto. A implementação de um sistema de reservas online, programas específicos para empresas, diversas opções de alojamento e uma loja de *merchandising* são aspetos que podem enriquecer significativamente a proposta do grupo, proporcionando uma experiência superior aos visitantes e uma gestão mais eficiente do parque.



FIGURA 3 - PARQUE AQUÁTICO DE AMARANTE

Parque Aquático de Amarante

A análise do site do Parque Aquático de Amarante oferece várias ideias que podem ser incorporadas no desenvolvimento do projeto CodeWave. A adoção de um design semelhante, moderno e apelativo, pode melhorar significativamente a experiência dos utilizadores. Além disso, a criação de uma secção de restauração detalhada, exibindo todos os menus disponíveis, pode fornecer informações valiosas aos visitantes.

Esta solução foi escolhida porque apresenta características fundamentais para o desenvolvimento do projeto. A inclusão de um design moderno e apelativo, juntamente com funcionalidades detalhadas como informações sobre restauração e uma galeria de fotos, pode enriquecer a proposta do grupo, proporcionando uma melhor experiência aos visitantes e uma gestão mais eficaz do parque.

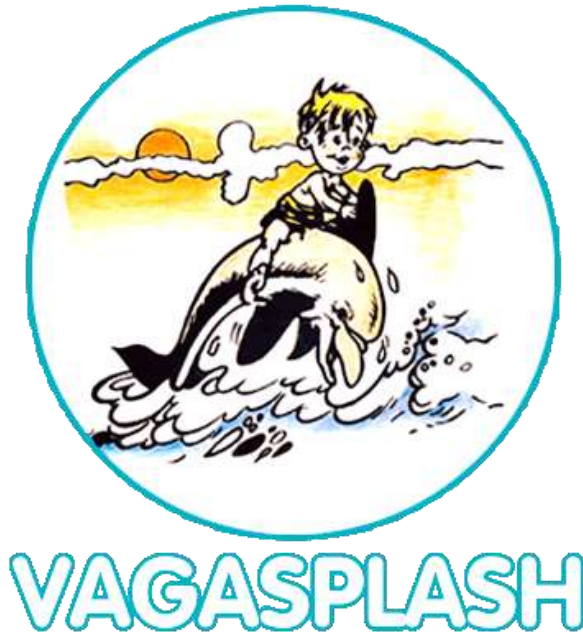


FIGURA 4 - VAGASPLASH

VagaSplash

A avaliação do website do VagaSplash sugere diversas áreas que podem ser exploradas no âmbito do projeto CodeWave. A integração de um design minimalista oferece potencial para melhorar tanto a estética quanto a usabilidade do site. Adicionalmente, a introdução de um sistema para organizar festas e eventos dentro do parque representa uma oportunidade significativa para atrair mais visitantes e impulsionar a receita do parque.

Esta solução contém características que o grupo considerou fundamentais para o desenvolvimento do projeto. A inclusão de um design simples e direto, juntamente com a possibilidade de organizar festas e eventos, pode enriquecer a proposta do grupo, proporcionando uma melhor experiência aos visitantes e uma gestão mais eficaz do parque.



FIGURA 5 - ZOOMARINE

ZooMarine

Após uma análise do website do ZooMarine, diversas oportunidades para o projeto CodeWave emergem. Destaca-se a possibilidade de implementar um mapa 2D semelhante ao presente no site, bem como a inclusão de um mapa detalhado dos transportes disponíveis para chegar até ao local. Além disso, a criação de uma aplicação móvel pode ampliar a acessibilidade e a interatividade para os visitantes.

Este parque aquático apresenta uma vasta gama de funcionalidades inovadoras, desde a organização de eventos até programas educacionais e de conservação. No entanto, o site peca por um excesso de elementos visuais desatualizados, contrastando com a modernidade de suas propostas. A oportunidade para o CodeWave reside na capacidade de harmonizar a criatividade e inovação presentes no parque com um design mais limpo e contemporâneo no seu website, garantindo assim uma experiência mais atrativa e coerente para os utilizadores.

3. Planeamento e Execução

No início do trabalho foi necessário utilizar os conhecimentos adquiridos em Programação e Tecnologias Web, relacionados com a modelação do projeto. Este ponto demonstra o trabalho necessário antes de se passar para a implementação do website em si.

3.1 Plano do projeto

Antes de se abordar o escalonamento das tarefas propriamente dito coligiram-se as tarefas, que se consideraram macro, a realizar. Com isto verificou-se que será necessário o seguinte:

- ♦ Documentação do projeto por meio de um relatório;
- ♦ Descrição do tema e dos utilizadores do sistema
- ♦ Identificação dos requisitos e funcionalidades do sistema;
- ♦ Levantamento do estado da arte;
- ♦ Descrição dos casos de utilização que fundamentam o que o sistema irá permitir fazer;
- ♦ Identificação e caracterização de todas as classes/interfaces, associações, atributos e métodos
- ♦ Produção do modelo de dados persistente;
- ♦ Implementação das funcionalidades do website;

De realçar também que as reuniões semanais, redação das atas, criação e gestão do grupo, bem como estrutura do relatório também se incluem como tarefas.

Todas estas atas podem ser vistas no **ANEXO E** de forma a se verificar o que foi discutido sobre o projeto ao longo do tempo.

3.2 Metodologia adotada

Neste projeto, foi utilizada a “metodologia em cascata” como metodologia de desenvolvimento de software. O modelo compreende quatro fases iniciais interligadas: Análise de Requisitos, Planeamento, Implementação e Manutenção. Estas etapas dependem sempre da conclusão bem-sucedida da fase anterior em que estas criam resultados que servem como base para a próxima fase. A simplicidade e a aplicabilidade direta são as principais vantagens deste modelo, favorecendo a documentação e o controlo de qualidade. No entanto, a sua inflexibilidade pode ser uma desvantagem, pois pode tornar-se difícil a adaptação a mudanças de requisitos ou no design ao longo do processo. Contudo, no contexto deste projeto, esses desafios não se manifestaram, e o modelo em cascata foi eficiente.

3.3 Calendarização das tarefas prevista

Na fase de planeamento de um projeto, a criação de um diagrama de Gantt é essencial para organizar e gerir as tarefas ao longo da sua execução. Este processo envolve determinar a melhor maneira de distribuir as tarefas no tempo, considerando a duração de cada uma, as relações de precedência entre elas e os prazos a serem cumpridos. Com base nesses critérios, foi elaborado um cronograma para o projeto, no qual estão especificadas as atividades previstas e o tempo estimado para a sua execução

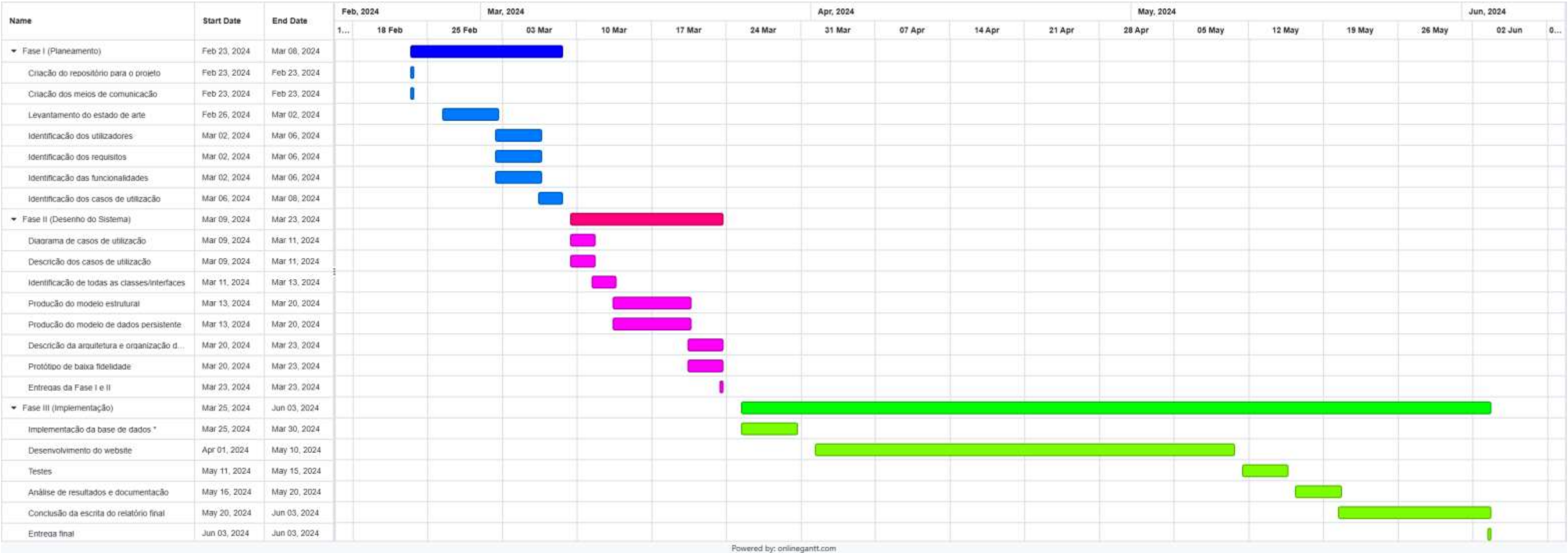


FIGURA 6 - CALENDARIZAÇÃO PREVISTA

3.4 Calendarização das tarefas executada

Em seguida, apresenta-se a execução real do trabalho através do seguinte diagrama de Gantt, que mostra como as tarefas foram efetivamente realizadas ao longo do projeto. Durante a implementação das tarefas, foi necessário fazer ajustes em relação ao cronograma original. A figura abaixo mostra a designação de cada tarefa, sua duração, e as datas de início e conclusão.

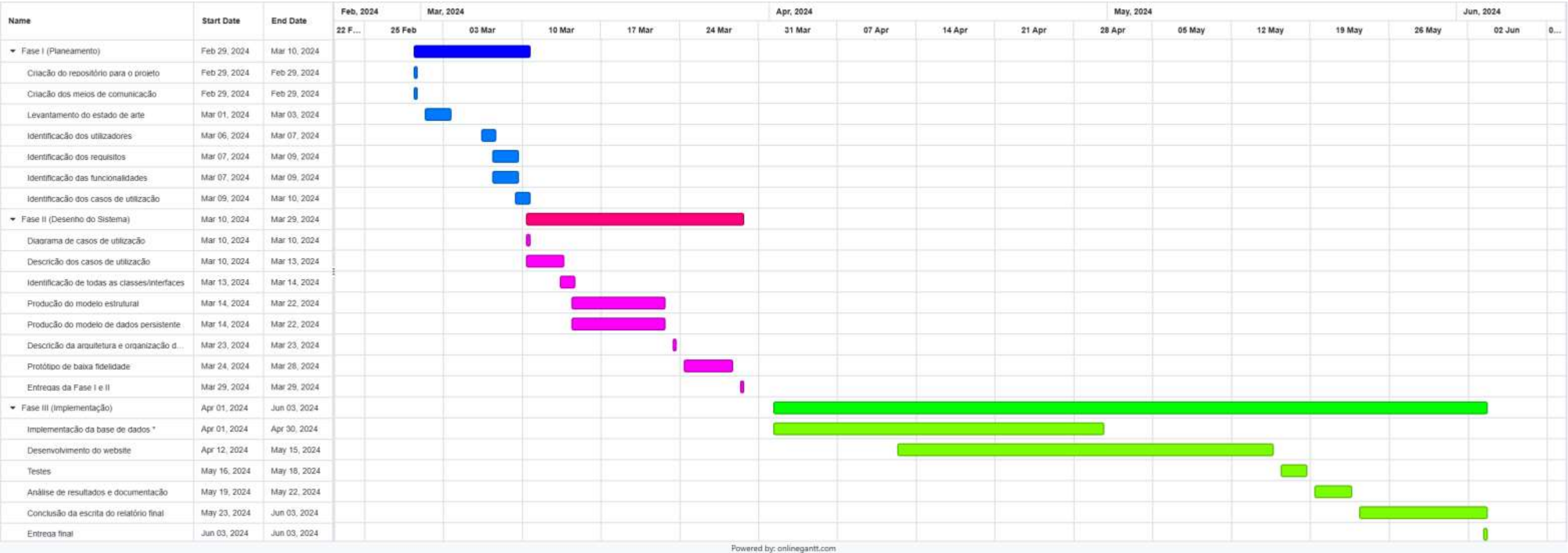


FIGURA 7 - CALENDARIZAÇÃO EXECUTADA

* A fase de implementação da base de dados estava inicialmente prevista para ser concluída em cinco dias. Contudo, enfrentámos desafios significativos que prolongaram este período para aproximadamente trinta dias. As principais dificuldades residiram no estabelecimento da conexão com a base de dados, na configuração de um túnel SSH e problemas com as credenciais de acesso. Apesar destes contratempos, foi possível manter o desenvolvimento do website em curso, recorrendo a uma base de dados temporária até que a situação fosse resolvida e a base de dados definitiva estivesse operacional.

3.5 Divisão de tarefas

Apresenta-se, em seguida, como as tarefas foram divididas pelos membros integrantes do grupo e como estas se sucederam ao longo do desenvolvimento do projeto.

#	Tarefa	Duração	Responsabilidade (%)					
1	Fase I (Planeamento)	10 dias						
1.1	Levantamento do estado da arte	3 dias	0	70	0	0	30	0
1.2	Identificação das funcionalidades do sistema	3 dias	20	20	20	20	20	0
1.3	Identificação dos utilizadores do sistema	2 dias	20	20	20	20	20	0
1.4	Identificação dos casos de utilização	2 dias	20	20	20	20	20	0
1.5	Identificação dos requisitos do sistema	3 dias	20	20	20	20	20	0
2	Fase II (Desenho do sistema)	20 dias						
2.1	Criação do diagrama de casos de utilização	1 dia	20	20	20	20	20	0
2.2	Descrição dos casos de utilização	4 dias	16.(6)	16.(6)	16.(6)	16.(6)	16.(6)	16.(6)
2.3	Identificação de todas as classes/interfaces	2 dias	16.(6)	16.(6)	16.(6)	16.(6)	16.(6)	16.(6)
2.4	Produção do modelo de dados persistente	9 dias	20	20	20	20	20	0
2.5	Produção do modelo estrutural	9 dias	20	20	20	20	20	0
2.6	Descrição da arquitetura e organização do website	1 dia	20	20	20	20	20	0
2.7	Protótipo de baixa fidelidade	5 dias	16.(6)	16.(6)	16.(6)	16.(6)	16.(6)	16.(6)
2.8	Elaboração do relatório intermédio	3 dias	16.(6)	16.(6)	16.(6)	16.(6)	16.(6)	16.(6)
3	Fase III (Implementação)	64 dias						
3.1	Correção de erros da Fase II	10 dias	20	20	20	20	20	0
3.2	Implementação da base de dados	30 dias	20	20	20	20	20	0
3.3	Implementação da aplicação web	34 dias	23.75	23.75	23.75	23.75	0 *	5
3.4	Testes e correção de eventuais falhas	3 dias	19	19	19	19	19	5
3.5	Análise de resultados e documentação	4 dias	16.(6)	16.(6)	16.(6)	16.(6)	16.(6)	16.(6)
3.6	Conclusão da elaboração do relatório final	12 dias	10	10	10	10	55 *	5

* Devido a dificuldades técnicas, o membro Miguel Pirré não conseguiu colaborar na parte de desenvolvimento da aplicação web, embora tenha acompanhado o progresso. Decidiu-se em grupo que ele iniciaria a elaboração do relatório final mais cedo do que o previsto, enquanto os demais membros continuariam com a implementação do website.

Bruno Migueis	Daniel Silva	Gabriel Cravo	Lucas Duarte	Miguel Pirré	Artiom Masulicov
---------------	--------------	---------------	--------------	--------------	------------------

FIGURA 8 - DIVISÃO DAS TAREFAS

4. Modelo de requisitos

No ponto que se segue são descritas as funcionalidades do sistema, isto é, especifica-se o que o sistema deve fazer, obrigatoriamente, e como o sistema se deve comportar, tendo em conta, por exemplo, a usabilidade e desempenho. Os subtópicos seguintes descrevem então os requisitos funcionais e não funcionais do sistema.

4.1 Requisitos funcionais

Os requisitos funcionais são definidos como as funcionalidades ou atividades que um sistema deve realizar. Representa o que o website faz, em termos de tarefas e serviços, através de cálculos, detalhes técnicos, manipulação de dados e de processamento e outras funcionalidades específicas que definem o que um sistema, idealmente, será capaz de realizar.

Em seguida está apresentada uma tabela [**Tabela 1** - Requisitos funcionais] que contém os requisitos funcionais alusivos ao desenvolvimento do sistema.

Funcionalidade do Utilizador: Indica que o requisito está relacionado a uma ação ou interação específica do utilizador.

Funcionalidade do Sistema: Refere-se a operações internas do sistema que não são diretamente perceptíveis pelo utilizador, mas que são necessárias para o funcionamento adequado da aplicação.

Integração: Diz respeito a requisitos que envolvem a interação com outros sistemas, módulos ou componentes.

Refª	Requisito funcional	Prioridade
RF.1	Comprar bilhetes	Elevada
RF.2	Gerir estacionamento	Baixa
RF.3	Gerir produtos	Elevada
RF.4	Gestão de equipamentos	Moderada
RF.5	Carrinho de compras	Elevada
RF.6	Visualização e interatividade do mapa do parque	Baixa
RF.7	Gestão de reservas para atividades específicas	Elevada
RF.8	Sistema de notificações e alertas	Baixa
RF.9	Remover <i>feedbacks</i>	Baixa
RF.10	Gerir cargos	Baixa
RF.11	Visualizar horários do parque aquático	Elevada
RF.12	Integração de API de meteorologia (<i>Open Weather API</i>)	Moderada
RF.13	Afluência do parque dia a dia	Moderada
RF.14	Gerir funcionários	Elevada
RF.15	Visualizar informações sobre segurança e regras do parque	Moderada
RF.16	Agendamento de eventos especiais	Elevada
RF.17	Gerir restauração	Moderada
RF.18	Autenticação	Elevada
RF.19	Validar bilhete	Elevada
RF.20	Gerir conta	Elevada
RF. 21	Chat com Integração de IA	Baixa
RF. 22	Alugar conveniências	Elevada

RF. 23	Gerar relatório de estatísticas do gerente	Moderada
RF. 24	Visualização do mapa do parque em realidade aumentada	Baixa
RF. 25	Comprar produto	Alta
RF. 26	Enviar <i>feedback</i>	Moderada
RF. 27	Gerir progressão de carreiras	Moderada
RF. 28	Visualizar menus da restauração	Baixa
RF. 29	Gerir horários do parque aquático	Alta
RF. 30	Visualizar galeria de fotos	Baixa
RF. 31	Contactar o parque aquático	Baixa
RF. 32	Alugar alojamento	Elevada
RF. 33	Ver notícias do parque	Baixa
RF. 34	Gerir notícias do parque	Baixa
RF. 35	Subscrever <i>newsletter</i>	Baixa
RF. 36	Gerir zonas	Baixa
RF. 37	Gerir categorias	Baixa

TABELA 1 - REQUISITOS FUNCIONAIS

4.2 Restrições e requisitos não funcionais

Os requisitos não funcionais devem conter elementos específicos, tais como a descrição da tarefa a ser executada pelo software, a origem do requisito e o seu utilizador, a relação da troca de informação entre o software e o utilizador, e se existirem, algumas restrições lógicas associadas à tarefa. Dentro dos requisitos não funcionais estão incluídos os requisitos de interface e de facilidade de uso, de desempenho, de segurança e integridade de dados, de interface com sistemas externos e ambientes de execução, entre outros.

Os requisitos não funcionais estão relacionados com os requisitos funcionais e indicam como o sistema/software deve ser feito e como deve funcionar, ou seja, são os critérios que qualificam os requisitos funcionais.

Foi-se elaborada uma tabela para as restrições [**Tabela 2**] e outra para os requisitos não funcionais [**Tabela 3**].

Restrição	Descrição
Ambiente de Implantação	O sistema deve ser implementado em servidores que atendam aos padrões de segurança exigidos para a gestão de vendas de um parque aquático
Segurança de Dados	O sistema deve proteger os dados, especialmente em relação às informações de clientes e transações financeiras.
Compatibilidade	O sistema deve ser compatível com os dispositivos utilizados no parque aquático, como terminais computadores.
Idioma	O sistema deve suportar o idioma português, pois o foco são os parques aquáticos portugueses.
Disponibilidade	O sistema deve estar disponível 24/7, com tempo de inatividade planeado mínimo para manutenção

TABELA 2 - RESTRIÇÕES

Refª	Requisito não funcional	Descrição
RNF.1	Desempenho	O sistema deve ser capaz de lidar com um grande número de pedidos simultâneos durante as horas de pico de pedidos
RNF.2	Escalabilidade	O sistema deve ser escalável para acomodar um aumento na procura durante eventos especiais ou épocas movimentadas
RNF.3	Usabilidade	A interface do utilizador deve ser intuitiva, permitindo que seja utilizada facilmente, mesmo em momentos de movimento intenso
RNF.4	Segurança	Medidas de segurança devem proteger os dados do sistema contra acesso não autorizado e garantir a integridade
RNF.5	Backup e Recuperação	Deve existir um plano de <i>backup</i> eficaz e um processo de recuperação de dados em caso de falhas
RNF.6	Integração	O sistema deve ser capaz de se integrar com sistemas de pagamento para aceitar diferentes métodos de pagamento
RNF.7	Manutenção e Suporte	Um plano de manutenção regular e suporte técnico deve estar disponível para garantir a estabilidade contínua do sistema
RNF.8	Tempo de Resposta	O sistema deve fornecer tempos de resposta rápidos, garantindo um atendimento eficiente aos clientes

TABELA 3 - REQUISITOS NÃO FUNCIONAIS

4.3 Requisitos de interface e facilidade de uso

Os requisitos de interface e facilidade de uso correspondem às expectativas e especificações, desenhados para assegurar que um produto, serviço, processo ou ambiente, seja fácil de utilizar.

Em seguida está apresentada uma tabela [**Tabela 4**] que contém os requisitos de interface e usabilidade, e respetivos requisitos funcionais, alusivos ao desenvolvimento do sistema.

Ref ^a	Requisito de interface e usabilidade	Req. funcionais relacionados
RInt.1	Website simples e intuitivo	
RInt.2	Implementar um design intuitivo para a interface do utilizador	
RInt.3	Garantir a consistência do design	
RInt.4	Dispor as informações do menu de forma lógica e simples	
RInt.5	Facilitar a movimentação dentro do website	RF.21 – Chat com integração com de IA

TABELA 4 - REQUISITOS DE INTERFACE E FACILIDADE DE USO

4.4 Requisitos de desempenho

Os requisitos de desempenho são requisitos criados à volta das necessidades da infraestrutura para garantir que funcione sem lentidão ou outros problemas.

Ref ^a	Requisito de desempenho	Req. funcionais relacionados
RDes.1	Website funcional e rápido ao carregar	
RDes.2	Ligação com um scanner de código de barras/QR	RF.24 – Visualização do mapa em Realidade Aumenta RF.1 – Comprar bilhete RF.19 – Validar Bilhete

TABELA 5 - REQUISITOS DE DESEMPENHO

4.5 Requisitos de segurança e integridade dos dados

Os requisitos de segurança e integridade de dados são requisitos que preservam todos os dados, seja por meio de criptografia, *hashing* certificados digitais, autenticações e controlo de acesso.

Ref ^a	Requisito de segurança, privacidade e integridade de dados	Req. funcionais relacionados
RSeg.1	Sistema de Backup	
RSeg.2	Validação de dados das contas de utilizador	RF.18 – Autenticação RF.20 – Gerir Conta
RSeg.3	Validação de dados de pagamento	
RSeg.4	O sistema não deve permitir que o funcionário ou administrador realize alterações sem estar autenticado.	RF.18 – Autenticação

TABELA 6 - REQUISITOS DE SEGURANÇA E INTEGRIDADE DOS DADOS

4.6 Requisitos de interface com sistemas externos e ambientes de execução

Os requisitos de interface com sistemas externos e com ambientes de execução podem ser colocados quer no produto, quer no processo, e são derivados do ambiente onde o sistema está a ser desenvolvido, baseando-se em informação do domínio de aplicação, na necessidade do sistema em interagir com outros sistemas, etc.

Em seguida está apresentada uma tabela que contém os requisitos de interface com sistemas externos e com ambientes de execução, e respetivos requisitos funcionais alusivos ao desenvolvimento do sistema.

Requisito	Descrição
R.I.1 – Scanner código de barras	É necessária conexão com um leitor de código de barras

TABELA 7 - REQUISITOS DE INTERFACE COM SISTEMAS EXTERNOS E AMBIENTES DE EXECUÇÃO

5. Normas e regulamentação específicas aplicáveis

Realizou-se uma pesquisa sobre todas as normas e regulamentações necessárias e aplicáveis, este passo é sempre importante para garantir a conformidade e qualidade da aplicação:

Faturação Eletrónica:

Norma: EN 16931: Esta norma estabelece o formato comum da fatura eletrónica na União Europeia (Norma Europeia EN).

Lei n.º 123/2019: Regula a utilização da fatura eletrónica, transpondo a Diretiva 2014/55/EU para a legislação portuguesa.

Proteção de Dados Pessoais:

Regulamento Geral de Proteção de Dados (RGPD): A Lei n.º 58/2019 assegura a aplicação do RGPD em Portugal. Os sistemas de gestão devem garantir a conformidade com a privacidade e a proteção de dados pessoais dos clientes.

Normas Fiscais e Contabilidade:

Código do Imposto sobre o Valor Acrescentado (CIVA): Estabelece as regras fiscais relacionadas com o IVA.

Código dos Regimes Contributivos do Sistema Previdencial de Segurança Social: Define as obrigações contributivas das empresas.

Normas para Software:

Regulamento (UE) 2016/679: Relativo à livre circulação de dados pessoais e à proteção da privacidade no setor das comunicações eletrónicas (Regulamento ePrivacy).

ISO/IEC 25010: Define critérios de qualidade para software, incluindo funcionalidade, desempenho, segurança e manutenibilidade.

Emissão de Documentos Eletrónicos:

Decreto-Lei n.º 28/2019: Regula a obrigação de utilização de programas informáticos de faturação e a emissão de documentos de transporte por via eletrónica.

Leis de Proteção ao Consumidor:

Regulamentos específicos de proteção ao consumidor podem aplicar-se à venda de produtos online, incluindo a divulgação clara de informações sobre os produtos, políticas de devolução e garantia, e a resolução de conflitos.

Normas de Acessibilidade Web: Assegurar que o website é acessível a todos os utilizadores, incluindo aqueles com deficiências, em conformidade com as normas de acessibilidade web, como as diretrizes do W3C para Conteúdo Acessível na Web (WCAG).

Legislação de Direitos de Autor e Propriedade Intelectual: Respeitar os direitos de autor e propriedade intelectual ao utilizar imagens, vídeos, música e outros conteúdos no website, garantindo que possua as licenças adequadas ou que o conteúdo seja de domínio público.

Regulamentos de Segurança Alimentar e Higiene: Se o website oferecer serviços de catering ou venda de alimentos e bebidas, é importante cumprir os regulamentos locais de segurança alimentar e higiene, garantindo a qualidade e segurança dos produtos.

Decreto Regulamentar Nº 5/97 | DR: Decreto lei que contém toda a informação necessária sobre infraestruturas necessárias ao funcionamento de um recinto de diversões aquáticas. Contém informação sobre limites de lotação, limites de profundidades dos tanques, proibições de certos materiais e todos os cuidados a ter nas determinadas diversões aquáticas.

Decreto Regulamentar Nº 65/97 | DR: Decreto lei que contém toda a informação necessária sobre regulamentos de licenças, manutenções e vistorias. Assim como sanções e coimas aplicáveis.

5.1 Requisitos de hardware

Os requisitos de hardware necessários para conseguir perfeitamente utilizar o **CodeWave** incluem: um servidor (para hospedar a aplicação), com acesso a uma conexão de Ethernet estável, estações de trabalho (para funcionários e o gerente) e um Arduino que tenha suporte de comunicação serial (no sentido de fazer a gestão das luzes na parte do gerente). Existem também requisitos de hardware opcionais como um leitor de código de barras com capacidade de ler códigos QR (para a leitura dos códigos QR dos bilhetes) e impressora (para imprimir as estatísticas do parque aquático).

Computador: 2 unidades (gerente e funcionário).

Componente	Requisitos recomendados
Processador	Intel core i5 8ª geração ou superior
Memória (RAM)	8 GB
Armazenamento	SSD de 500 Gb
Placa Gráfica	Integrada
Conectividade	Placa de Rede Ethernet (porta Ethernet), Bluetooth e WiFi(opcional)
Sistema Operativo	Windows 10 ou superior
Portas e Conectores	Portas USB para periféricos

TABELA 8 - REQUISITOS DE HARDWARE COMPUTADOR

Computador (servidor): 1 unidade.

Componente	Requisitos recomendados
Processador	Intel core i5 de 8ª geração ou superior
Memória (RAM)	16 GB
Armazenamento	SSD de 500 Gb
Rede	Ethernet Gigabit (conexão de rede estável e rápida)

TABELA 9 - REQUISITOS DE HARDWARE SERVIDOR

6. Diagrama de Arquitetura do Sistema

A imagem abaixo [Figura 9 - Diagrama de Arquitetura do Sistema], ilustra a arquitetura de uma aplicação web integrada com um sistema de hardware em Arduino. Esta arquitetura compreende a interação do utilizador através de uma interface web (frontend), a gestão dos dados (backend), e a comunicação com um dispositivo Arduino para a recolha e envio de dados.

A aplicação utiliza várias tecnologias, como GitHub Pages para hospedar os arquivos frontend, um servidor web para processar requisições, scripts PHP armazenados num servidor de ficheiros FTP, e uma base de dados PostgreSQL para armazenar informações. Além disso, a comunicação entre o Arduino e o computador gestor é realizada via USB (Comunicação Serial), com um script Python responsável pelo processamento dos dados entre a Base de Dados e o Arduino.

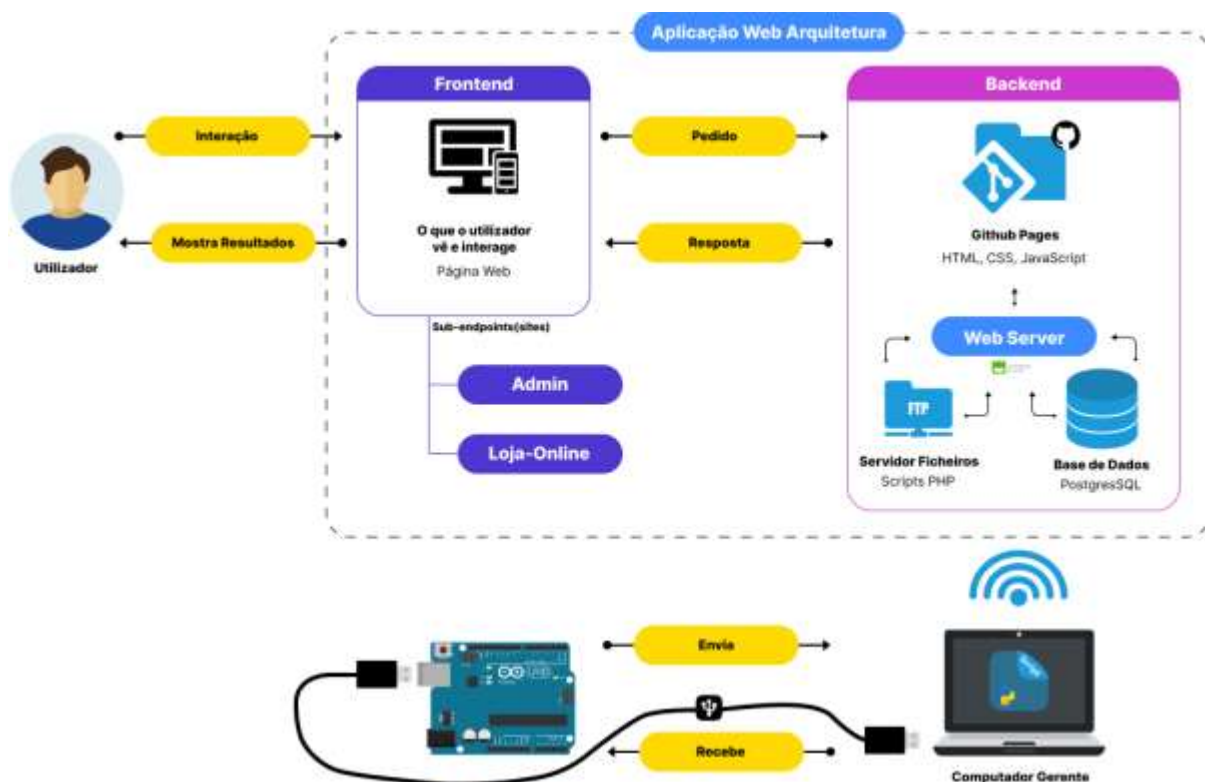


FIGURA 9 - DIAGRAMA DE ARQUITETURA DO SISTEMA

7. Diagrama de atividades

No desenvolvimento de sistemas, a modelagem visual é uma ferramenta crucial para a compreensão e comunicação de processos complexos. Um dos diagramas mais utilizados para esse fim é o Diagrama de Atividades, uma representação gráfica que descreve o fluxo de atividades dentro de um sistema. Originário da linguagem de modelagem unificada (UML), o diagrama de atividades é essencial para detalhar o comportamento dinâmico do sistema, mostrando como as atividades são sequenciadas e como os fluxos de controlo e de dados são organizados.

De seguida ilustram-se dois diagramas de atividades, o de login para o gerente e funcionários e o de login para utilizadores.

7.1 Login para o gerente e funcionários

Na figura seguinte representa-se o diagrama de atividades de login para o gerente e funcionários. No mesmo podem-se verificar os diferentes caminhos e ações tendo em conta as decisões dos diferentes nós existentes.

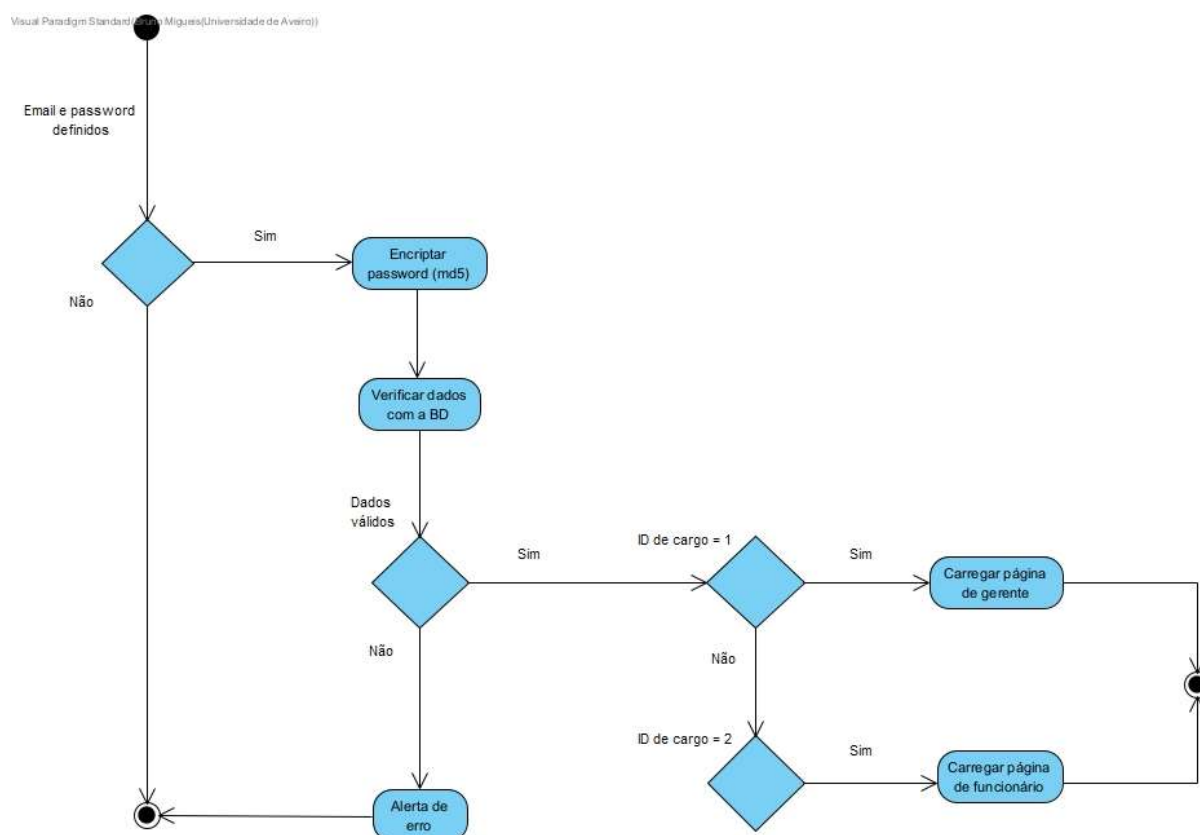


FIGURA 10 - DIAGRAMA DE ATIVIDADES DE LOGIN PARA O GERENTE E FUNCIONÁRIOS

7.2 Login para utilizadores

Na figura seguinte representa-se o diagrama de atividades de login para os utilizadores. No mesmo podem-se verificar os diferentes caminhos e ações tendo em conta as decisões dos diferentes nós existentes.

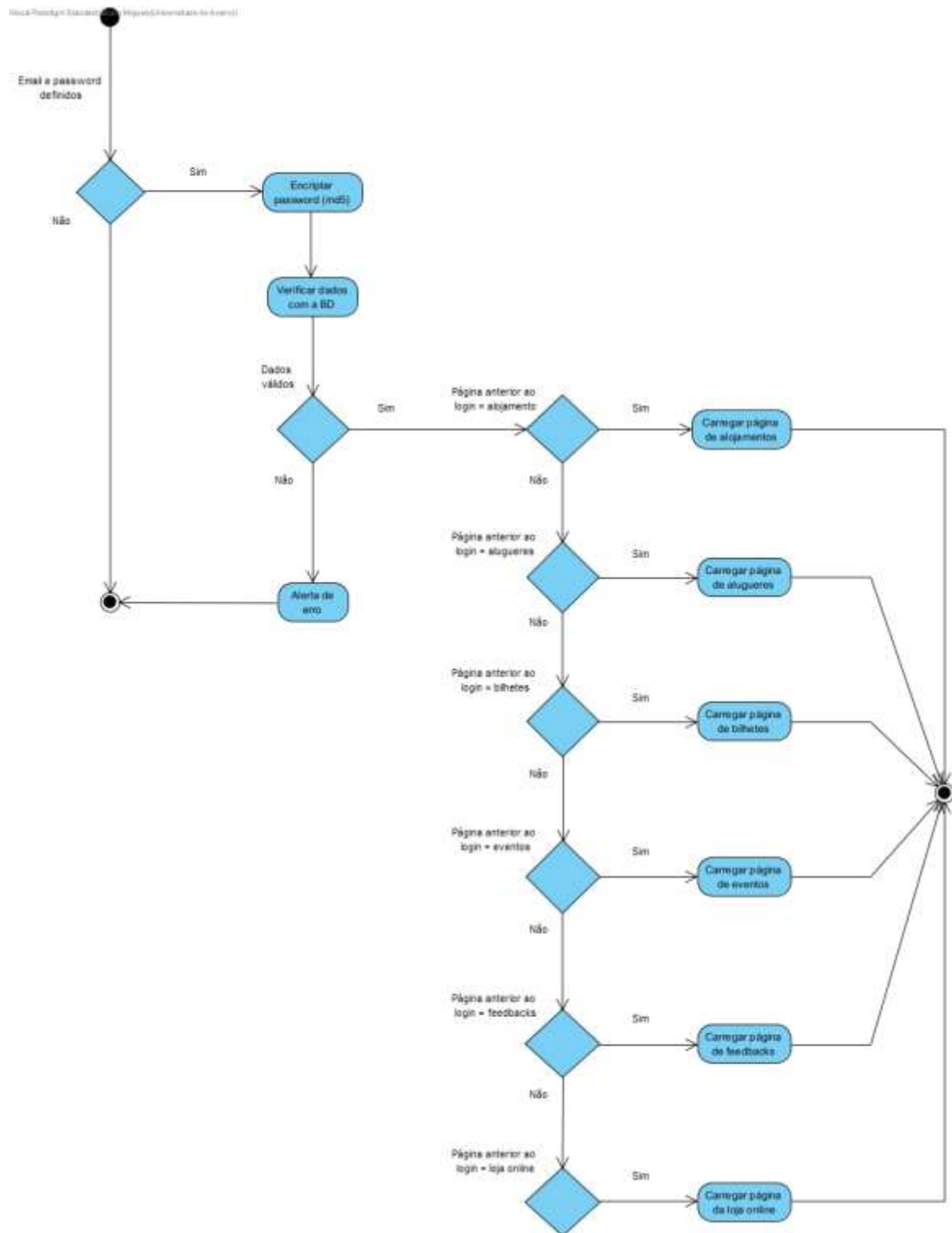


FIGURA 11 - DIAGRAMA DE ATIVIDADES DE LOGIN PARA OS UTILIZADORES

8. Casos de Utilização

Os diagramas de casos de utilização tipicamente contêm uma lista de ações ou passos que definem as interações entre um ator e o sistema a fim de atingir um dado objetivo. Em relação ao nosso website em questão, há um conjunto de atores que interagirão com esta, que neste caso, consistirá no gerente e funcionários. Todos terão ao dispor funcionalidades que a plataforma fornece, tais como: gerir bilhetes, gerir alugueres ou gerir alojamentos entre outros, onde cada uma dessas ações se podem, tecnicamente, transformar em casos de utilização. Os subtópicos seguintes descrevem os atores e os casos de utilização que integram o sistema, assim como o diagrama de casos de uso. Detalham-se os casos de utilização referindo o seguinte: atores, requisitos funcionais e grau de prioridade associados; pré-condições (o estado do sistema antes do caso de uso), pós-condições (o estado do sistema depois de concluído o caso de uso), fluxo básico (série de afirmações declarativas que listam os passos do caso de uso), fluxo alternativo (alternativas ao caminho básico) e ainda a sua finalidade (descrição sumária).

8.1 Atores

O ator especifica um papel executado por um utilizador ou outro sistema que interage com o sistema. O ator deve ser externo ao sistema e deve ter associações exclusivamente para casos de uso, componentes ou classes, podendo herdar o papel de outro. Na tabela abaixo apresentada [Tabela 10] é possível descrever os atores que interagem com o sistema.

Actor	Descrição
Funcionário	Pessoa capaz de realizar vendas e fazer validação de bilhetes
Gerente	Pessoa capaz de gerir totalmente o website e os seus dados
Utilizador	Pessoa que visita o site podendo comprar bilhetes, objetos relacionados com o tema e alugar material, espaço no local ou alugar alojamento.

TABELA 10 - ATORES DO SISTEMA

8.2 Casos de utilização

No seguinte diagrama evidenciam-se os casos de uso (contêm uma lista de ações ou passos que definem as interações entre um ator e o sistema).

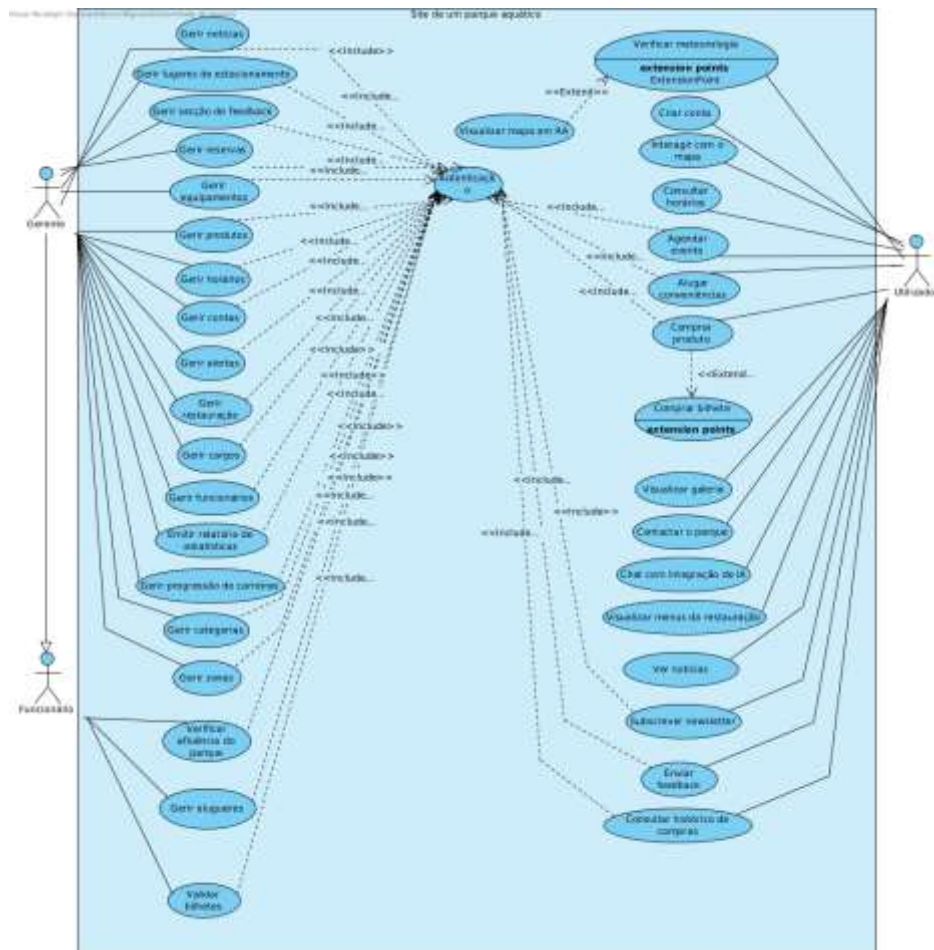


FIGURA 12 - CASOS DE UTILIZAÇÃO

Para além do diagrama anterior, abaixo evidencia-se o diagrama de casos de uso referente à loja online, no qual se destacam os casos de uso (contêm uma lista de ações ou passos que definem as interações entre um ator e o sistema na loja online).

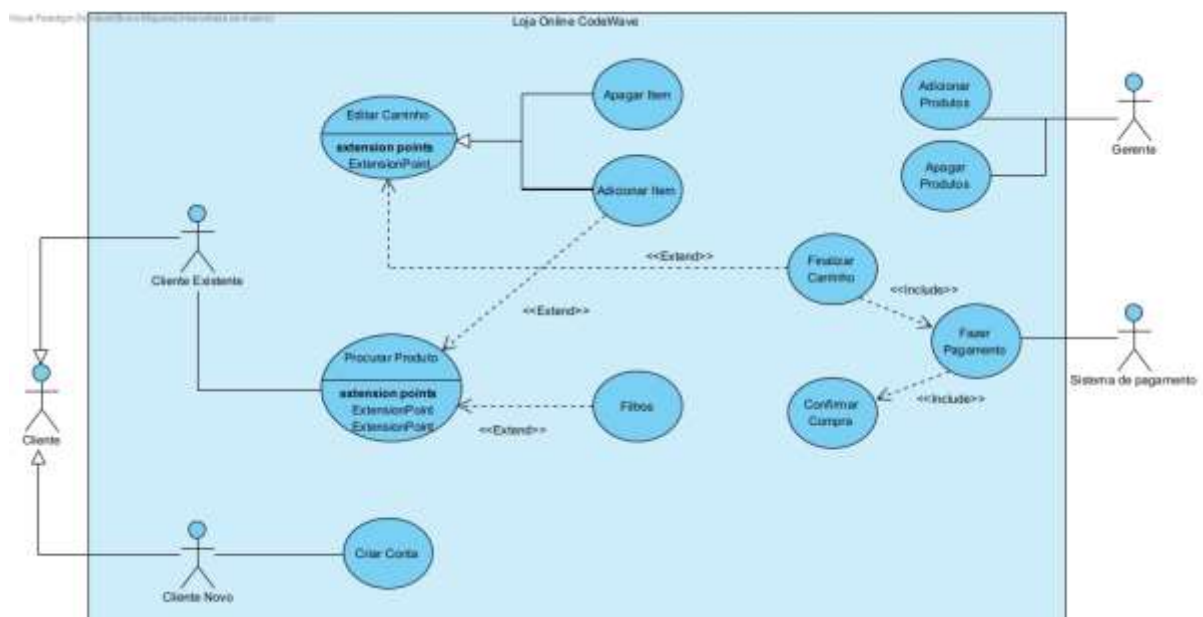


FIGURA 13 - DIAGRAMA DE CASOS DE UTILIZAÇÃO DA LOJA ONLINE

Nome:	Gerir lugares de estacionamento
Atores:	Funcionário
Prioridade (1/3):	1
Finalidade:	Gestão de lugares de estacionamento do parque.
Requisitos funcionais:	RF.2 – Gerir Estacionamento
Pré-condições:	O website (<i>dashboard</i>) já têm dois sensores para controlar os lugares disponíveis.
Sumário:	O funcionário verifica quantos lugares livres tem no estacionamento.
Sequência típica dos eventos	
<div style="display: flex; justify-content: space-between;"> <div>Ações dos atores</div> <div>Respostas do sistema</div> </div>	
O funcionário gere o número de lugares disponíveis no parque.	O sistema regista e mostra o número de lugares disponíveis em tempo real.

TABELA 11 CASO DE UTILIZAÇÃO GERIR LUGARES DE ESTACIONAMENTO

Nome:	Gerir secção de <i>feedback</i>
Atores:	Gerente
Prioridade (1/3):	1
Finalidade:	Verificar o <i>feedback</i> dado por parte dos clientes com o objetivo de fazer melhorias ao parque.
Requisitos funcionais:	RF.9 – Remover <i>feedbacks</i>
Pré-condições:	Necessário haver pelo menos um <i>feedback</i> dado.
Sumário:	O gerente acede à página de controlo do parque e verifica os <i>feedbacks</i> dados.
Sequência típica dos eventos	
<div style="display: flex; justify-content: space-between;"> <div>Ações dos atores</div> <div>Respostas do sistema</div> </div>	
O gerente encontra o um <i>feedback</i> inválido ou incoerente e elimina-o.	O sistema elimina o <i>feedback</i> do utilizador.

TABELA 12 CASO DE UTILIZAÇÃO GERIR SECÇÃO DE FEEDBACK

Nome:	Gerir reservas
Atores:	Funcionário
Prioridade (1/3):	3
Finalidade:	Gerir uma reserva de eventos.
Requisitos funcionais:	RF.7 – Gestão de reservas para atividades específicas
Pré-condições:	Um utilizador deve fazer uma reserva.
Sumário:	O utilizador faz a reserva, e o funcionário verifica o seu estado.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador faz a reserva de um evento ou espaço numa determinada data.	O sistema apresenta um formulário para fazer a reserva.

TABELA 13 CASO DE UTILIZAÇÃO GERIR RESERVAS

Nome:	Gerir equipamentos
Atores:	Gerente
Prioridade (1/3):	3
Finalidade:	Gestão dos equipamentos presentes no parque.
Requisitos funcionais:	RF.4 – Gestão de equipamentos
Pré-condições:	É necessário que haja sensores instalados.
Sumário:	O gerente pode controlar os equipamentos que estão no parque.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O gerente liga as luzes todas do parque através da interface de utilizador.	O sistema liga as luzes presentes no parque.

TABELA 14 CASO DE UTILIZAÇÃO GERIR EQUIPAMENTOS

Nome:	Gerir produtos
Atores:	Gerente
Prioridade (1/3):	3
Finalidade:	Gestão de todos os produtos.
Requisitos funcionais:	RF.3 – Gerir produtos
Pré-condições:	É necessário que haja produtos para venda.
Sumário:	O utilizador retira alguns produtos que já não estão disponíveis e adiciona novos.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O gerente adiciona o produto espreguiçadeira a um preço de 3 euros.	O sistema regista o novo produto com o preço atribuído.

TABELA 15 CASO DE UTILIZAÇÃO GERIR PRODUTOS

Nome:	Gerir contas
Atores:	Gerente
Prioridade (1/3):	3
Finalidade:	Criação, edição ou remoção de contas de funcionários.
Requisitos funcionais:	RF.20 – Gerir conta
Pré-condições:	Tem de existir uma conta gerente.
Sumário:	O gerente pode criar, editar ou apagar contas.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O gerente remove uma conta de um funcionário.	O sistema remove a conta do funcionário.

TABELA 16 CASO DE UTILIZAÇÃO GERIR CONTAS

Nome:	Gerir alertas
Atores:	Gerente
Prioridade (1/3):	1
Finalidade:	Gestão de notificações do parque aquático, enviadas para os utilizadores através de e-mail ou SMS.
Requisitos funcionais:	RF.8 – Sistema de notificações e alertas
Pré-condições:	O sistema deve estar em funcionamento. O gerente deve estar autenticado no sistema.
Sumário:	O caso de utilização “Gerir Alertas” permite ao gerente controlar as notificações enviadas aos utilizadores do parque aquático. Essas notificações podem incluir informações sobre eventos especiais, alterações nos horários de funcionamento, condições meteorológicas adversas ou outras atualizações relevantes.
Sequência típica dos eventos	
<div style="display: flex; justify-content: space-between;"> Ações dos atores Respostas do sistema </div>	
<p>Configurar Alertas: O gerente define os tipos de alertas que deseja enviar (por exemplo, eventos, promoções, avisos de encerramento).</p> <p>Definir Destinatários: O gerente especifica os grupos de utilizadores que devem receber os alertas (por exemplo, visitantes, membros, funcionários).</p> <p>Criar Notificações: O gerente redige as mensagens de notificação, incluindo o conteúdo e o formato (e-mail ou SMS).</p> <p>Agendar Envio: O gerente programa o envio das notificações para momentos específicos (por exemplo, antes de um evento, durante uma emergência).</p>	<p>O sistema envia as notificações aos destinatários conforme configurado pelo gerente.</p> <p>Regista o histórico de envio.</p> <p>Notifica o gerente sobre quaisquer problemas de entrega ou falhas.</p>

TABELA 17 CASO DE UTILIZAÇÃO GERIR ALERTAS

Nome:	Gerir alugueres
Atores:	Gerente
Prioridade (1/3):	3
Finalidade:	Criação, edição ou remoção de alugueres feitos por clientes.
Requisitos funcionais:	RF.22 – Alugar conveniências
Pré-condições:	Ter conta de gerente.
Sumário:	O gerente seria capaz de criar editar ou remover alugueres caso seja necessário.
Sequência típica dos eventos	
<div style="display: flex; justify-content: space-between;"> Ações dos atores Respostas do sistema </div>	
O gerente edita um aluguer de um utilizador.	O sistema altera o aluguer do cliente.

TABELA 18 CASO DE UTILIZAÇÃO GERIR ALUGUERES

Nome:	Validar bilhete				
Atores:	Funcionário				
Prioridade (1/3):	3				
Finalidade:	Validação dos bilhetes pelo funcionário.				
Requisitos funcionais:	RF.1 – Comprar bilhete, RF.19 – Validar bilhete				
Pré-condições:	É necessária uma conta de gerente ou de funcionário.				
Sumário:	O funcionário seria capaz de verificar se o bilhete existe e que pertence à pessoa. Validando a entrada no parque.				
<p align="center">Sequência típica dos eventos</p> <table> <tr> <th>Ações dos atores</th><th>Respostas do sistema</th></tr> <tr> <td>O funcionário introduz o código do bilhete.</td><td>Sistema apresenta os dados relacionados a esse bilhete.</td></tr> </table>		Ações dos atores	Respostas do sistema	O funcionário introduz o código do bilhete.	Sistema apresenta os dados relacionados a esse bilhete.
Ações dos atores	Respostas do sistema				
O funcionário introduz o código do bilhete.	Sistema apresenta os dados relacionados a esse bilhete.				

TABELA 19 CASO DE UTILIZAÇÃO VALIDAR BILHETE

Nome:	Autenticação				
Atores:	Utilizador, Funcionário, Gerente				
Prioridade (1/3):	3				
Finalidade:	Saber quem é a pessoa que está a utilizar o site.				
Requisitos funcionais:	RF.18 – Autenticação, RF.20 – Gerir conta				
Pré-condições:					
Sumário:	O utilizador deve fazer a autenticação caso queira usufruir de mais funcionalidades do site.				
<p align="center">Sequência típica dos eventos</p> <table> <tr> <th>Ações dos atores</th><th>Respostas do sistema</th></tr> <tr> <td>O utilizador faz a autenticação no website</td><td>O sistema concede ao cliente funcionalidades, tais como a compra ou o aluguer de material.</td></tr> </table>		Ações dos atores	Respostas do sistema	O utilizador faz a autenticação no website	O sistema concede ao cliente funcionalidades, tais como a compra ou o aluguer de material.
Ações dos atores	Respostas do sistema				
O utilizador faz a autenticação no website	O sistema concede ao cliente funcionalidades, tais como a compra ou o aluguer de material.				

TABELA 20 CASO DE UTILIZAÇÃO AUTENTICAÇÃO

Nome:	Criar conta
Atores:	Utilizador
Prioridade (1/3):	3
Finalidade:	Permitir que os utilizadores criem uma conta no sistema do parque aquático.
Requisitos funcionais:	RF.20 – Gerir conta
Pré-condições:	O sistema deve estar em funcionamento. O utilizador deve ter acesso à interface de registo.
Sumário:	O caso de utilização “Criar Conta” possibilita que os utilizadores se registem no sistema do parque aquático. Ao criar uma conta, os utilizadores podem aceder a funcionalidades específicas, como comprar bilhetes, receber atualizações e personalizar as suas preferências.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
<p>Acesso à Página de Registo: O utilizador acede à página de registo através do website do parque aquático.</p> <p>Preenchimento de Dados: O utilizador introduz as informações necessárias para criar a conta, como nome, endereço de e-mail, palavra-passe, etc.</p> <p>Validação de Dados: O sistema verifica se os dados fornecidos pelo utilizador são válidos (por exemplo, se o endereço de e-mail é único e se a palavra-passe atende aos requisitos de segurança).</p> <p>Criação da Conta: Se os dados forem válidos, o sistema cria uma conta para o utilizador e atribui um identificador único.</p> <p>Notificação de Sucesso: O sistema informa o utilizador que a conta foi criada com sucesso e fornece instruções adicionais (por exemplo, verificar o e-mail para ativar a conta).</p>	<p>O sistema regista os detalhes da nova conta na base de dados.</p> <p>Envia um e-mail de confirmação ou ativação para o utilizador, se necessário.</p> <p>Redireciona o utilizador para a página de início de sessão após a criação da conta.</p>

TABELA 21 CASO DE UTILIZAÇÃO CRIAR CONTA

Nome:	Verificar meteorologia
Atores:	Utilizador
Prioridade (1/3):	1
Finalidade:	O utilizador verifica a meteorologia.
Requisitos funcionais:	RF.12 – Integração de API de meteorologia (<i>Open Weather API</i>)
Pré-condições:	O utilizador deve aceder ao website.
Sumário:	O utilizador tem acesso a meteorologia de um certo dia.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador acede o website para verificar a meteorologia.	O sistema exibe a meteorologia do dia selecionado pelo cliente.

TABELA 22 CASO DE UTILIZAÇÃO VERIFICAR METEOROLOGIA

Nome:	Interagir com o mapa
Atores:	Utilizador
Prioridade (1/3):	2
Finalidade:	O utilizador interage com o mapa do parque.
Requisitos funcionais:	RF.6 – Visualização e interatividade do mapa do parque
Pré-condições:	O utilizador deve aceder ao website.
Sumário:	O utilizador interage com as atrações.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador acede o mapa interativo do parque aquático no website.	O sistema carrega e exhibe o mapa interativo do parque aquático com as atrações.

TABELA 23 CASO DE UTILIZAÇÃO INTERAGIR COM O MAPA

Nome:	Consultar horários
Atores:	Todos os utilizadores
Prioridade (1/3):	1
Finalidade:	Permitir que os utilizadores obtenham informações atualizadas sobre os horários de funcionamento do parque aquático.
Requisitos funcionais:	RF.11 – Visualizar horários do parque aquático, RF.21 – Chat de integração com AI
Pré-condições:	O sistema deve estar em funcionamento. Os horários de funcionamento do parque aquático devem estar definidos no sistema.
Sumário:	O caso de utilização “Consultar Horários” permite aos utilizadores visualizarem os horários de abertura, encerramento e quaisquer alterações especiais (por exemplo, feriados, eventos) do parque aquático. Essas informações são essenciais para que os visitantes possam planear as suas visitas de acordo com os horários disponíveis.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
Acesso à Página de Horários: O utilizador navega para a secção de horários no website do parque aquático.	O sistema exhibe os horários de funcionamento de forma clara e organizada. Fornece informações adicionais, como horários de espetáculos, aulas de natação ou outras atividades programadas.

TABELA 24 CASO DE UTILIZAÇÃO CONSULTAR HORÁRIOS

Nome:	Agendar evento
Atores:	Utilizador
Prioridade (1/3):	3
Finalidade:	O utilizador agenda um evento.
Requisitos funcionais:	RF.16 – Agendamento de eventos especiais; RF.7 – Gestão de reservas para atividades específicas
Pré-condições:	O utilizador deve ter uma conta.
Sumário:	O utilizador pode agendar um dia para fazer eventos especiais.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador seleciona a opção de agendar um evento	O sistema exibe um formulário para o utilizador preencher com os detalhes do evento (data, hora, participantes).

TABELA 25 CASO DE UTILIZAÇÃO AGENDAR EVENTO

Nome:	Alugar conveniências
Atores:	Utilizador
Prioridade (1/3):	3
Finalidade:	Permitir que os utilizadores aluguem equipamentos e serviços adicionais para melhorar a sua experiência no parque aquático.
Requisitos funcionais:	RF.22 – Alugar conveniência
Pré-condições:	O sistema deve estar em funcionamento. Os produtos e serviços disponíveis para aluguer devem estar registados no sistema.
Sumário:	O caso de utilização “Alugar Conveniências” permite aos utilizadores acederem a uma variedade de conveniências e serviços mediante pagamento adicional.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
<p>Seleção de Conveniências: O utilizador escolhe os produtos ou serviços que deseja alugar.</p> <p>Verificação de Disponibilidade: O sistema verifica se os produtos selecionados estão disponíveis na data e hora desejadas.</p> <p>Efetuar Pagamento: O utilizador efetua o pagamento pelo aluguer das conveniências.</p> <p>Receber Confirmação: O sistema confirma o aluguer e fornece informações relevantes (por exemplo, local de recolha, horários de devolução).</p>	<p>O sistema regista os alugueres efetuados pelos utilizadores.</p> <p>Gera recibos ou comprovativos de pagamento.</p> <p>Notifica os utilizadores sobre quaisquer alterações ou cancelamentos relacionados com os alugueres.</p>

TABELA 26 CASO DE UTILIZAÇÃO ALUGAR CONVENIÊNCIAS

Nome:	Comprar produto
Atores:	Utilizador
Prioridade (1/3):	2
Finalidade:	Permitir que os utilizadores comprem produtos, como bilhetes, lembranças, protetores solares, brinquedos ou outros produtos disponíveis na loja do parque aquático.
Requisitos funcionais:	RF.1 – Comprar bilhetes; RF.25 - Comprar produto
Pré-condições:	O sistema deve estar em funcionamento. Os produtos devem estar registados no sistema, com informações como preço, descrição e disponibilidade.
Sumário:	O caso de utilização “Comprar Produto” permite aos utilizadores explorarem a loja do parque aquático e adquirirem os produtos que desejam. Esses produtos podem variar desde pequenos souvenirs até produtos essenciais para a sua estadia no parque.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
<p>Exploração da Loja: O utilizador navega pela loja do parque aquático, visualizando os produtos disponíveis.</p> <p>Seleção de Produto: O utilizador escolhe um ou mais produtos que deseja comprar.</p> <p>Adição ao Carrinho: O sistema permite que o cliente adicione os produtos selecionados ao seu carrinho de compras virtual.</p> <p>Verificação de Preço e Quantidade: O utilizador verifica o preço total dos produtos no carrinho e a quantidade disponível.</p> <p>Efetuar Pagamento: O utilizador efetua o pagamento pelos produtos selecionados.</p> <p>Receber Confirmação: O sistema confirma a compra, gera um recibo e fornece informações sobre a entrega ou recolha dos produtos.</p>	<p>O sistema regista as compras efetuadas pelos utilizadores.</p> <p>Atualiza automaticamente o inventário dos produtos após cada compra.</p> <p>Notifica o cliente sobre quaisquer problemas de pagamento ou disponibilidade.</p>

TABELA 27 CASO DE UTILIZAÇÃO COMPRAR PRODUTO

Nome:	Comprar bilhete
Atores:	Utilizador
Prioridade (1/3):	3
Finalidade:	O utilizador compra bilhetes para entrar no parque aquático.
Requisitos funcionais:	RF.1 - Comprar bilhetes, RF.19 – Validar bilhete, RF.21 – Chat com integração AI
Pré-condições:	O utilizador deve ter uma conta no website.
Sumário:	O utilizador compra 1 bilhete para o dia que vai ao parque.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador compra um bilhete online para uma determinada data.	O sistema apresenta os bilhetes disponíveis e um formulário.

TABELA 28 CASO DE UTILIZAÇÃO COMPRAR BILHETE

Nome:	Verificar afluência do parque
Atores:	Utilizador
Prioridade (1/3):	2
Finalidade:	O utilizador verifica afluência do parque.
Requisitos funcionais:	RF.2 -Gerir estacionamento, RF.13 – Afluência do parque dia a dia
Pré-condições:	O utilizador deve aceder ao website.
Sumário:	O utilizador verifica afluência do parque em tempo real.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador seleciona a opção de verificar a afluência do parque.	O sistema exhibe os dados da afluência do parque em tempo real.

TABELA 29 CASO DE UTILIZAÇÃO VERIFICAR AFLUÊNCIA DO PARQUE

Nome:	Consultar histórico de compras
Atores:	Utilizador
Prioridade (1/3):	2
Finalidade:	O utilizador verifica as suas compras antigas.
Requisitos funcionais:	RF.20 – Gerir conta
Pré-condições:	O utilizador deve ter uma conta, e tem de ter compras realizadas na loja.
Sumário:	Após uma compra na loja, o cliente verifica o seu histórico de compras.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador consulta o seu histórico de compras.	O sistema responde apresentando o histórico de compras do utilizador.

TABELA 30 CASO DE UTILIZAÇÃO CONSULTAR HISTÓRICO DE COMPRAS

Nome:	Gerir restauração
Atores:	Gerente
Prioridade (1/3):	2
Finalidade:	Criar, editar, remover artigos do menu do bar.
Requisitos funcionais:	RF.17 – Gerir restauração. RF. 28 – Visualizar menu da restauração
Pré-condições:	Ter conta de gerente.
Sumário:	O gerente cria um menu do bar que vai ser apresentado no site em que pode editar os artigos que estão apresentados.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O gerente remove um artigo do menu do bar.	O sistema remove o artigo do sistema.

TABELA 31 CASO DE UTILIZAÇÃO GERIR RESTAURAÇÃO

Nome:	Editar carrinho de compras
Atores:	Utilizador
Prioridade (1/3):	3
Finalidade:	Visualizar, remover ou alterar a quantidade dos produtos do carrinho para efetuar uma compra.
Requisitos funcionais:	RF.5 – Carrinho de compras
Pré-condições:	Ter conta de utilizador.
Sumário:	Utilizador visualiza os produtos que adicionou ao carrinho e no mesmo pode remove ou altera a quantidade dos produtos adicionados.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador remove o produto do carrinho.	O sistema remove o produto do carrinho.

TABELA 32 CASO DE UTILIZAÇÃO EDITAR CARRINHO DE COMPRAS

Nome:	Procurar produto
Atores:	Utilizador
Prioridade (1/3):	2
Finalidade:	Procurar o produto da loja online através de uma barra de pesquisa.
Requisitos funcionais:	RF.25 – Comprar produto
Pré-condições:	Estar a página de todos os produtos da loja online.
Sumário:	Utilizador procura o nome do produto na loja e ao fazer filtra os produtos que contenham as letras do que escreveu ou falou.
Sequência típica dos eventos	
<div> Ações dos atores </div> <div> Respostas do sistema </div>	
O cliente começa a escrever “bo”.	O sistema apresenta os produtos que contenham “bo” neste caso bóia flamingo, bóia unicórnio e bounce ball.

TABELA 33 CASO DE UTILIZAÇÃO PROCURAR PRODUTO

Nome:	Enviar <i>feedback</i>
Atores:	Utilizador
Prioridade (1/3):	2
Finalidade:	O utilizador envia <i>feedback</i> .
Requisitos funcionais:	RF.26 – Enviar <i>feedback</i>
Pré-condições:	O utilizador deve ter uma conta no website.
Sumário:	O utilizador pode enviar <i>feedback</i> após preencher o formulário.
Sequência típica dos eventos	
<div> Ações dos atores </div> <div> Respostas do sistema </div>	
O utilizador preenche o formulário do <i>feedback</i> e clica no botão para enviar o <i>feedback</i> .	O sistema regista o <i>feedback</i> na base de dados e exibe uma mensagem de confirmação de envio.

TABELA 34 CASO DE UTILIZAÇÃO ENVIAR FEEDBACK

Nome:	Gerir progressão de carreiras				
Atores:	Gerente				
Prioridade (1/3):	2				
Finalidade:	O gerente gere a progressão de carreiras dos funcionários, como por exemplo, o cargo, salário, etc.				
Requisitos funcionais:	RF.27 – Gerir progressão de carreiras				
Pré-condições:	Ter conta de gerente.				
Sumário:	O gerente acede à secção de progressão, onde pode visualizar, editar e atualizar as informações dos funcionários.				
Sequência típica dos eventos					
<table> <tr> <th>Ações dos atores</th><th>Respostas do sistema</th></tr> <tr> <td>O gerente seleciona um funcionário para gerir a progressão de carreira.</td><td>O sistema exibe o perfil do funcionário com os detalhes da sua carreira, e guarda os campos atualizados.</td></tr> </table>		Ações dos atores	Respostas do sistema	O gerente seleciona um funcionário para gerir a progressão de carreira.	O sistema exibe o perfil do funcionário com os detalhes da sua carreira, e guarda os campos atualizados.
Ações dos atores	Respostas do sistema				
O gerente seleciona um funcionário para gerir a progressão de carreira.	O sistema exibe o perfil do funcionário com os detalhes da sua carreira, e guarda os campos atualizados.				

TABELA 35 CASO DE UTILIZAÇÃO GERIR PROGRESSÃO DE CARREIRAS

Nome:	Visualizar menus da restauração						
Atores:	Utilizador						
Prioridade (1/3):	2						
Finalidade:	Permitir que o utilizador veja os menus disponíveis nos restaurantes/bares do parque.						
Requisitos funcionais:	RF.28 - Visualizar menus da restauração						
Pré-condições:	Nenhuma.						
Sumário:	O utilizador visualiza os menus dos restaurantes através da aplicação web.						
Sequência típica dos eventos							
<table> <tr> <th>Ações dos atores</th><th>Respostas do sistema</th></tr> <tr> <td>O utilizador navega até à página “Restauração” através do menu da aplicação;</td><td>O sistema apresenta ao utilizador todos os produtos de todos os restaurantes/bares do parque;</td></tr> <tr> <td>O utilizador seleciona a opção para visualizar o menu de um restaurante/bar específico;</td><td>O sistema apresenta o menu do estabelecimento selecionado;</td></tr> </table>		Ações dos atores	Respostas do sistema	O utilizador navega até à página “Restauração” através do menu da aplicação;	O sistema apresenta ao utilizador todos os produtos de todos os restaurantes/bares do parque;	O utilizador seleciona a opção para visualizar o menu de um restaurante/bar específico;	O sistema apresenta o menu do estabelecimento selecionado;
Ações dos atores	Respostas do sistema						
O utilizador navega até à página “Restauração” através do menu da aplicação;	O sistema apresenta ao utilizador todos os produtos de todos os restaurantes/bares do parque;						
O utilizador seleciona a opção para visualizar o menu de um restaurante/bar específico;	O sistema apresenta o menu do estabelecimento selecionado;						

TABELA 36 CASO DE UTILIZAÇÃO VISUALIZAR MENUS DA RESTAURAÇÃO

Nome:	Gerir horários
Atores:	Gerente
Prioridade (1/3):	3
Finalidade:	Facilitar a gestão e atualização dos horários de funcionamento do parque aquático.
Requisitos funcionais:	RF.29 – Gerir horários do parque aquático
Pré-condições:	O gerente deve estar autenticado no sistema numa conta com permissões de administrador.
Sumário:	O gerente pode definir e atualizar os horários de funcionamento do parque aquático. Isto inclui a hora de abertura e hora de fecho de cada dia e dias em que esteja encerrado, tanto para o público geral como para escolas.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O gerente acede ao sistema de gestão; O gerente seleciona a opção de gerir horários; O gerente altera e guarda os novos horários;	O sistema solicita autenticação como administrador; O sistema apresenta os horários atuais; O sistema atualiza as informações conforme as alterações.

TABELA 37 CASO DE UTILIZAÇÃO GERIR HORÁRIOS

Nome:	Visualizar galeria
Atores:	Utilizador
Prioridade (1/3):	1
Finalidade:	Permitir que o utilizador visualize e interaja com uma galeria de fotos.
Requisitos funcionais:	RF.30 – Visualizar galeria de fotos
Pré-condições:	Nenhuma.
Sumário:	O utilizador navega por uma galeria de fotos, visualiza imagens em tamanho completo e interage com elas através de zoom e navegação para ficar a conhecer melhor o parque.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador abre o website e navega até à galeria de fotos; O utilizador escolhe uma foto para visualizar; O utilizador faz zoom na foto; O utilizador filtra as imagens pela secção que pretende visualizar;	O sistema exhibe todas as fotos da galeria; O sistema mostra a foto em tamanho completo; O sistema ajusta a imagem para mostrar mais detalhes; O sistema exhibe apenas as fotos da secção selecionada.

TABELA 38 CASO DE UTILIZAÇÃO VISUALIZAR GALERIA

Nome:	Contactar o parque
Atores:	Utilizador
Prioridade (1/3):	2
Finalidade:	Entrar em contacto com o parque aquático para obter informações ou resolver questões.
Requisitos funcionais:	RF.31 – Contactar o parque aquático
Pré-condições:	Estar na página de contacto do parque aquático.
Sumário:	O utilizador entra em contacto com o parque aquático para obter informações ou esclarecer dúvidas.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador preenche o formulário de contacto do parque.	O sistema processa o contacto do utilizador e envia para o email do parque o pedido/dúvida.

TABELA 39 CASO DE UTILIZAÇÃO CONTACTAR O PARQUE

Nome:	Alugar alojamento
Atores:	Utilizador
Prioridade (1/3):	2
Finalidade:	Poder alugar um alojamento.
Requisitos funcionais:	RF.32 – Alugar alojamento
Pré-condições:	Ter uma conta registada.
Sumário:	O utilizador pode alugar Bungalows disponíveis no website.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O utilizador aluga um Bungalow XL durante 3 dias.	O sistema retira da Base de dados a quantidade de Bungalow XL para aqueles dias e gera um código de levantamento de alojamento.

TABELA 40 CASO DE UTILIZAÇÃO ALUGAR ALOJAMENTO

Nome:	Ver notícias
Atores:	Utilizador
Prioridade (1/3):	2
Finalidade:	O utilizador poderá ver as notícias do parque.
Requisitos funcionais:	RF.33 – Ver notícias do parque
Pré-condições:	Nenhuma.
Sumário:	O utilizador poderá saber de notícias interessantes que aconteceram ou vão acontecer no website.
Sequência típica dos eventos <div> Ações dos atores Respostas do sistema </div>	
O utilizador selecciona uma notícia. O sistema apresenta a notícia ao utilizador.	

TABELA 41 CASO DE UTILIZAÇÃO VER NOTÍCIAS

Nome:	Gerir notícias
Atores:	Gerente
Prioridade (1/3):	2
Finalidade:	O gerente poderá gerir as notícias do parque.
Requisitos funcionais:	RF.34 – Gerir notícias do parque
Pré-condições:	Estar a página de todos os produtos da loja online.
Sumário:	O gerente poderá editar, apagar ou criar notícias interessantes para os utilizadores.
Sequência típica dos eventos <div> Ações dos atores Respostas do sistema </div>	
O gerente cria uma notícia sobre um evento no parque. O sistema apresenta para o utilizador a notícia nova.	

TABELA 42 CASO DE UTILIZAÇÃO GERIR NOTÍCIAS

Nome:	Subscrever <i>newsletter</i>				
Atores:	Utilizador				
Prioridade (1/3):	1				
Finalidade:	Subscrever a <i>newsletter</i> do site para receber informações sobre novas notícias, promoções e eventos.				
Requisitos funcionais:	RF.35 – Subscrever <i>newsletter</i>				
Pré-condições:	O utilizador deve ter sessão iniciada, para poder subscrever.				
Sumário:	O utilizador procura a opção de subscrever a <i>newsletter</i> .				
<p style="text-align: center;">Sequência típica dos eventos</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 50%; text-align: center;">Ações dos atores</th> <th style="width: 50%; text-align: center;">Respostas do sistema</th> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">O utilizador clica no botão de subscrever.</td> <td style="border: 1px solid black; padding: 5px;">O sistema associa a conta do utilizador para receber a <i>newsletter</i>.</td> </tr> </table>		Ações dos atores	Respostas do sistema	O utilizador clica no botão de subscrever.	O sistema associa a conta do utilizador para receber a <i>newsletter</i> .
Ações dos atores	Respostas do sistema				
O utilizador clica no botão de subscrever.	O sistema associa a conta do utilizador para receber a <i>newsletter</i> .				

TABELA 43 CASO DE UTILIZAÇÃO SUBSCREVER NEWSLETTER

Nome:	Gerir zonas				
Atores:	Gerente				
Prioridade (1/3):	2				
Finalidade:	Poder editar as zonas do parque para os eventos.				
Requisitos funcionais:	RF.36 – Gerir zonas				
Pré-condições:	Estar autenticado como gerente.				
Sumário:	O gerente pode editar, criar ou apagar novas zonas.				
<p style="text-align: center;">Sequência típica dos eventos</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <th style="width: 50%; text-align: center;">Ações dos atores</th> <th style="width: 50%; text-align: center;">Respostas do sistema</th> </tr> <tr> <td style="border: 1px solid black; padding: 5px;">O gerente cria uma zona.</td> <td style="border: 1px solid black; padding: 5px;">O sistema adiciona uma nova zona no sistema.</td> </tr> </table>		Ações dos atores	Respostas do sistema	O gerente cria uma zona.	O sistema adiciona uma nova zona no sistema.
Ações dos atores	Respostas do sistema				
O gerente cria uma zona.	O sistema adiciona uma nova zona no sistema.				

TABELA 44 CASO DE UTILIZAÇÃO GERIR ZONAS

Nome:	Gerir categorias
Atores:	Gerente
Prioridade (1/3):	3
Finalidade:	Poder editar as categorias dos produtos.
Requisitos funcionais:	RF.37 – Gerir categorias
Pré-condições:	Estar autenticado como gerente.
Sumário:	Poder editar as categorias dos produtos para uma melhor organização.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O gerente edita a categoria de um produto.	O sistema altera a categoria do produto para a nova.

TABELA 45 CASO DE UTILIZAÇÃO GERIR CATEGORIAS

Nome:	Gerir funcionários
Atores:	Gerente
Prioridade (1/3):	3
Finalidade:	Possibilidade de gerir os funcionários do parque.
Requisitos funcionais:	RF.14 – Gerir funcionários
Pré-condições:	Estar autenticado na página de gerente.
Sumário:	O gerente pode editar os funcionários do parque.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O gerente elimina um funcionário do parque.	O sistema elimina o funcionário do sistema.

TABELA 46 CASO DE UTILIZAÇÃO GERIR FUNCIONÁRIOS

Nome:	Gerir cargos
Atores:	Gerente
Prioridade (1/3):	3
Finalidade:	Gerir os cargos dos funcionários.
Requisitos funcionais:	RF.10 – Gerir cargos
Pré-condições:	Haver conta criada como funcionário.
Sumário:	O gerente pode editar o cargo de um funcionário.
Sequência típica dos eventos	
Ações dos atores	Respostas do sistema
O gerente muda o cargo de um funcionário para gerente.	O sistema atualiza a conta do funcionário para gerente.

TABELA 47 CASO DE UTILIZAÇÃO GERIR CARGOS

Nome:	Emitir relatório de estatísticas
Atores:	Gerente
Prioridade (1/3):	2
Finalidade:	Receber um relatório de estatísticas.
Requisitos funcionais:	RF.23 – Gerar relatório de estatísticas do gerente
Pré-condições:	Estar o autenticado numa conta de gerente.
Sumário:	O gerente na página principal da sua conta pode gerar um relatório de estatísticas.
Sequência típica dos eventos	
<div> <div>Ações dos atores</div> <div>Respostas do sistema</div> </div>	
<div> <div>O gerente seleciona gerar relatório de estatística.</div> <div>O sistema descarrega em PDF o relatório de estatísticas.</div> </div>	

TABELA 48 CASO DE UTILIZAÇÃO EMITIR RELATÓRIO DE ESTATÍSTICAS

Nome:	Visualizar mapa em RA
Atores:	Utilizador
Prioridade (1/3):	1
Finalidade:	O utilizador visualiza o mapa do parque em realidade aumentada.
Requisitos funcionais:	RF.24 – Visualização do mapa do parque em realidade aumentada
Pré-condições:	O utilizador deve ter aplicação da Adobe Aero no telemóvel.
Sumário:	Utilizador acede há página do mapa interativo e faz scan com o telemóvel do código QR, que abre o mapa em RA.
Sequência típica dos eventos	
<div> <div>Ações dos atores</div> <div>Respostas do sistema</div> </div>	
<div> <div>O utilizador faz scan com o telemóvel do código QR</div> <div>O sistema carrega o mapa em RA a partir do telemóvel do utilizador.</div> </div>	

TABELA 49 CASO DE UTILIZAÇÃO VISUALIZAR MAPA EM RA

Nome:	Chat com integração de IA	
Atores:	Utilizador	
Prioridade (1/3):	1	
Finalidade:	Obter informações de forma rápida através de um chat IA.	
Requisitos funcionais:	RF.21 – Chat com integração de IA	
Pré-condições:	Nenhuma.	
Sumário:	Utilizador comunica com um <i>chatbot</i> de inteligência artificial através de mensagens de texto de modo a obter informações sobre o parque.	
Sequência típica dos eventos		
Ações dos atores		Respostas do sistema
O utilizador pergunta ao <i>chatbot</i> quais os horários do parque.		O sistema responde ao utilizador com os horários do parque.

TABELA 50 CASO DE UTILIZAÇÃO CHAT COM INTEGRAÇÃO DE IA

8.3 Cobertura de requisitos

Uma tabela de cobertura de requisitos permite-nos avaliar de que forma os requisitos funcionais de um sistema são cobertos pelos casos de uso, garantindo assim que todos os requisitos funcionais são considerados. Na tabela abaixo, esses mesmos requisitos são especificados e interligados com o caso de uso correspondente.

Requisitos																																					
	RF.1	RF.2	RF.3	RF.4	RF.5	RF.6	RF.7	RF.8	RF.9	RF.10	RF.11	RF.12	RF.13	RF.14	RF.16	RF.17	RF.18	RF.19	RF.20	RF.21	RF.22	RF.23	RF.24	RF.25	RF.26	RF.27	RF.28	RF.29	RF.30	RF.31	RF.32	RF.33	RF.34	RF.35	RF.36	RF.37	
Casos de uso																																					
Gerir lugares de estacionamento		X																																			
Gerir secção de feedback									X																												
Gerir reservas							X																														
Gerir equipamentos				X																																	
Gerir produtos			X																																		
Gerir horários																													X								
Gerir contas																			X																		
Gerir alertas								X																													
Gerir alugueres																					X																
Validar bilhetes	X																		X																		
Autenticação																		X		X																	
Criar conta																				X																	
Verificar meteorologia												X																									
Interagir com o mapa						X																															
Consultar horários											X										X																
Agendar eventos							X								X																						
Alugar conveniências																					X																
Comprar produto																								X													
Comprar bilhete	X																		X		X																
Verificar afluência do parque		X											X																								
Enviar feedback																								X													
Consultar histórico de compras																			X																		
Gerir restauração																X											X										
Emitir relatório de estatísticas																						X															
Gerir progressão de carreira																										X											

TABELA 51 - COBERTURA DE REQUISITOS

9. Prototipagem baixa-fidelidade

Pode-se definir um protótipo, neste contexto, como a expressão de uma intenção de design a fim de obter uma simulação da interação final entre o utilizador com a interface. Já a fidelidade, como o nível de detalhe e realismo do protótipo. Assim, um protótipo de baixa-fidelidade é uma forma fácil e rápida de traduzir conceitos complexos de desenho em algo tangível e passível de ser testado. Estes têm como principal objetivo verificar e testar a funcionalidade ao invés da aparência visual do produto mostrando os maiores pedaços de conteúdo, posicionamento dos elementos e o alinhamento dos mesmos. Numa fase inicial do projeto, após o levantamento de funcionalidades e requisitos do sistema, chegou-se a um protótipo.



FIGURA 14 - HOMEPAGE

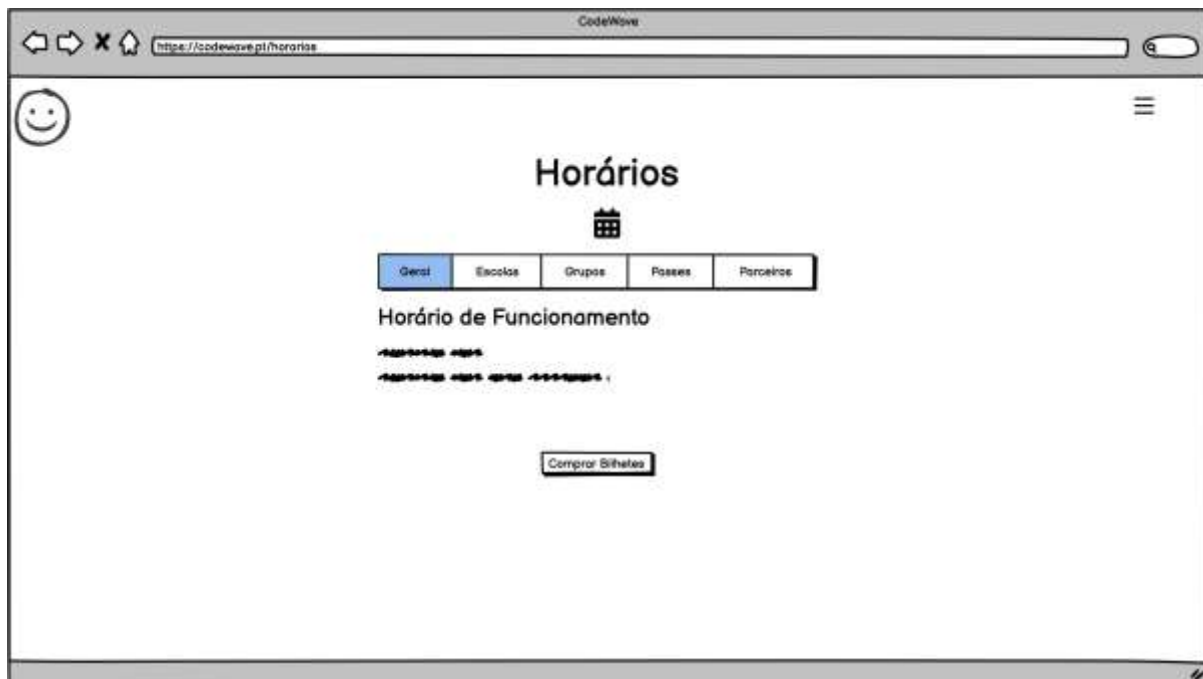


FIGURA 15 - HORÁRIOS

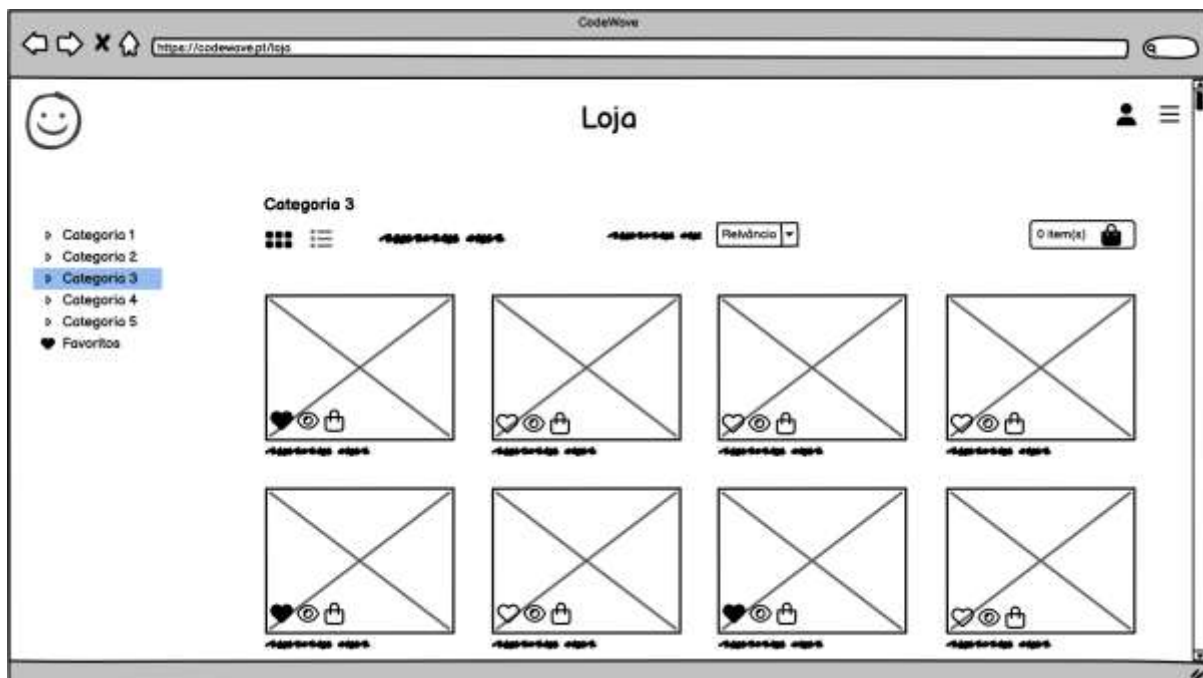


FIGURA 16 - LOJA ONLINE

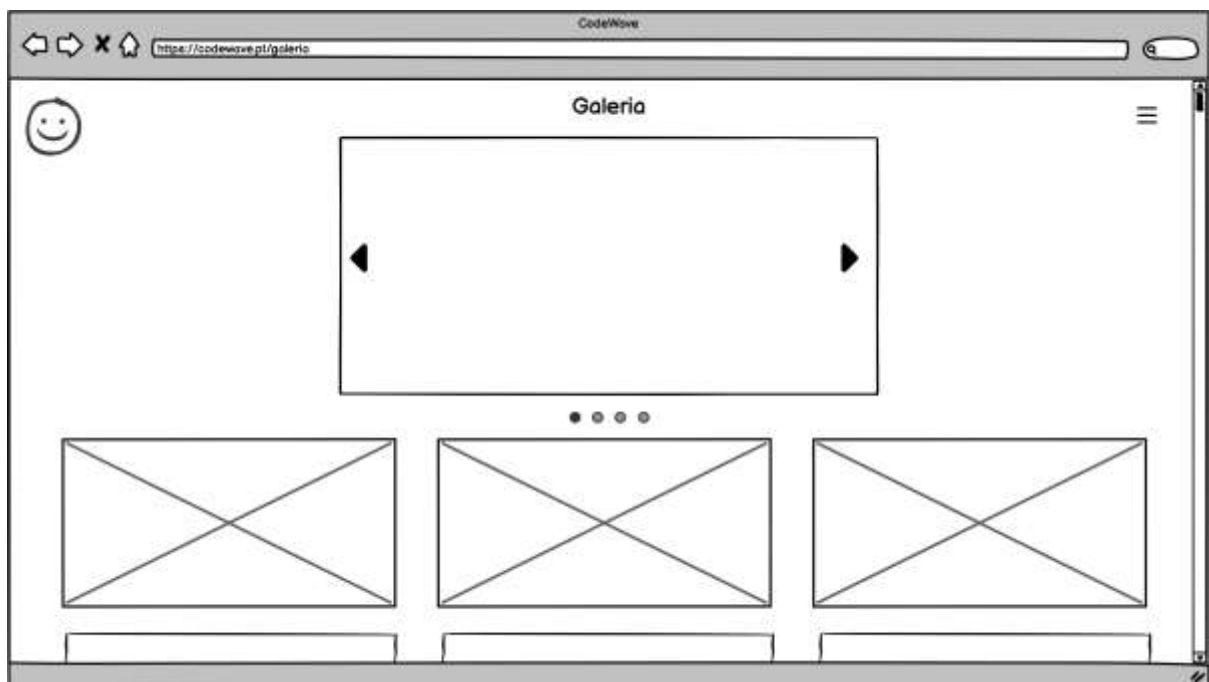


FIGURA 17 - GALERIA

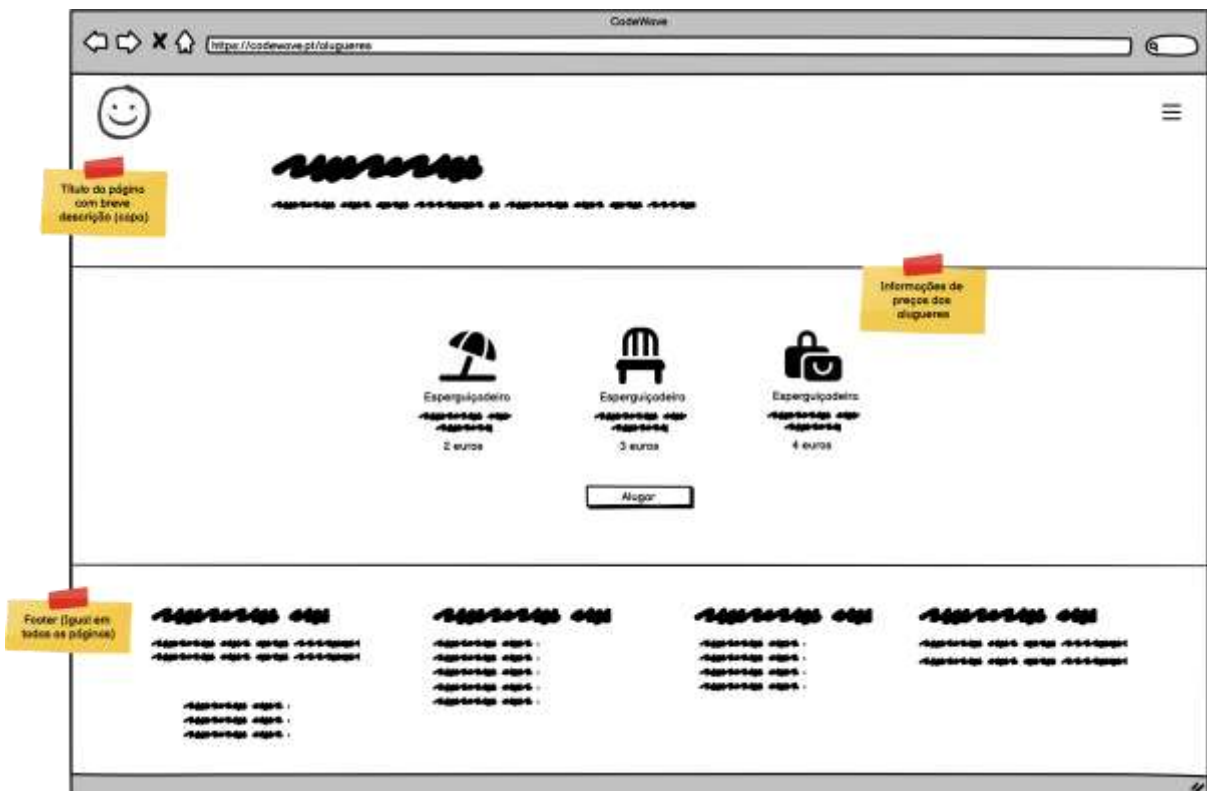


FIGURA 18 - ALUGUERES

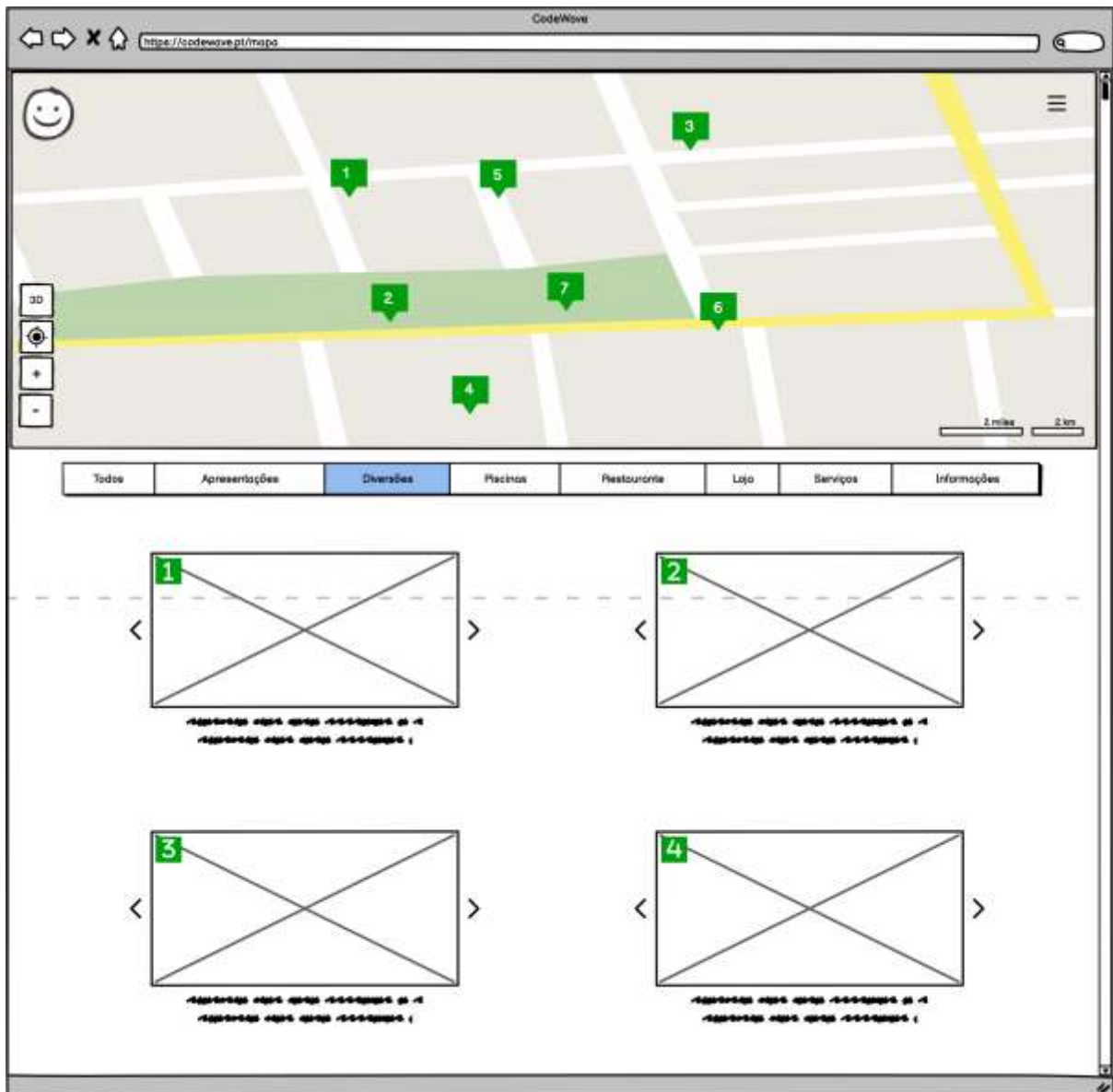


FIGURA 19 - MAPA



FIGURA 20 - DASHBOARD GERENTE

10. Análise de tecnologias

Considerando as tarefas definidas e a temática deste trabalho na qual consiste em desenvolver uma aplicação web, com acesso a uma base de dados, fez-se a análise de um conjunto de ferramentas e soluções para tornar isto possível:

- ♦ **pgAdmin 4:** É uma ferramenta visual abrangente que permite criar, editar e gerir bases de dados PostgreSQL. Este software também fornece funcionalidades de modelação de dados, desenvolvimento SQL, ferramentas de administração, gestão de utilizadores, backups, entre outras.

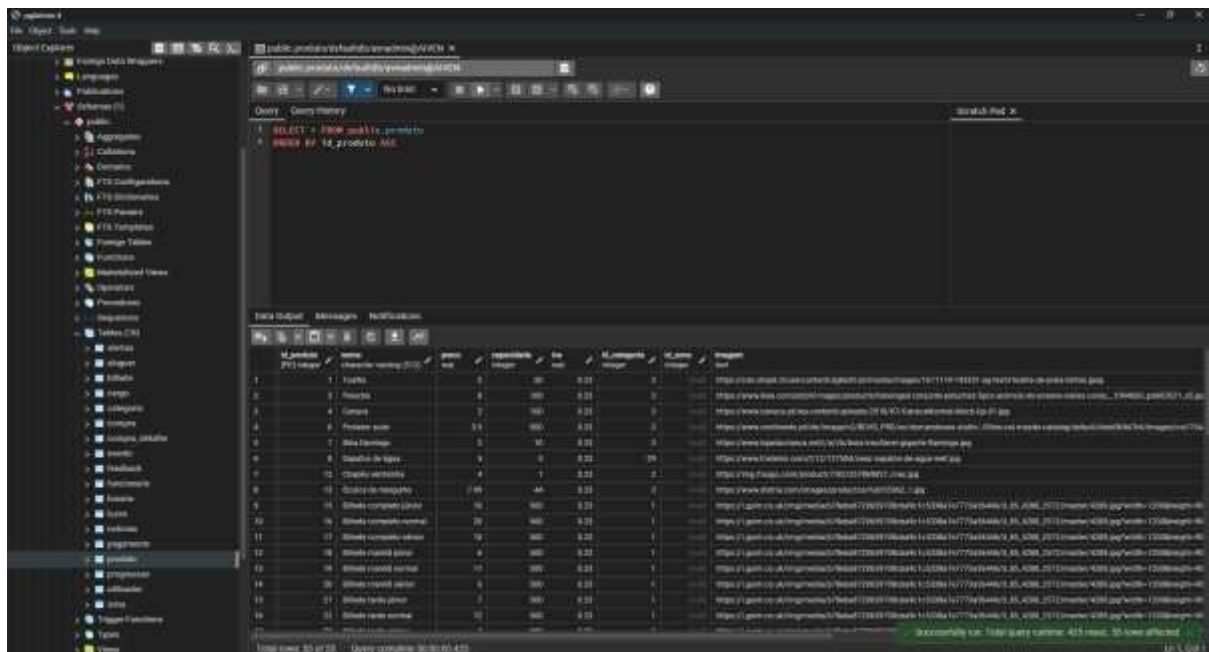


FIGURA 21 - PGADMIN 4

- ♦ **Git (repositório):** Software *open-source* e gratuito concebido para lidar com tudo desde projetos pequenos a muito grandes. Este é amplamente utilizado por desenvolvedores de software para gerir o código fonte dos seus projetos.
- ♦ **GitHub:** O GitHub é uma plataforma de hospedagem de código fonte que permite aos utilizadores armazenarem e partilharem os seus repositórios Git online. Esta ferramenta fornece uma interface gráfica do utilizador para o Git e adiciona recurso de colaboração, como rastreamento de problemas, solicitações de pull e administração de projetos. Utiliza no nosso projeto, visto que o grupo já está familiarizado com ela devido a projetos anteriores.

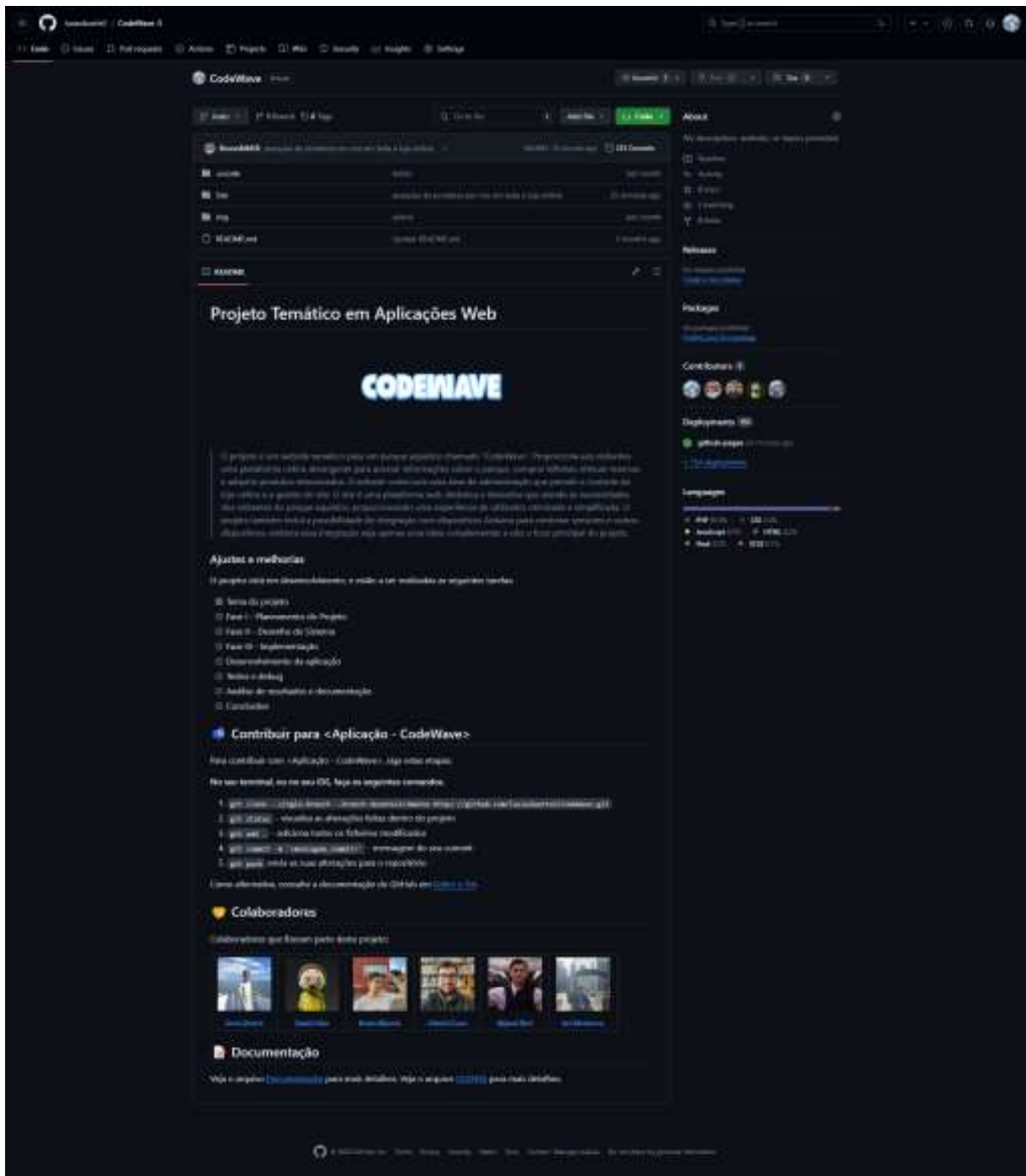


FIGURA 22 - GITHUB DO PROJETO CODEWAVE

- ♦ **Visual Studio Code (VS Code):** É um ambiente de desenvolvimento integrado (IDE) leve e flexível, desenvolvido pela Microsoft. Foi utilizado no projeto devido à familiaridade do grupo com a ferramenta e por oferecer funcionalidades que facilitam a utilização do Git. O VSCode suporta uma ampla gama de linguagens de programação e possui integração nativa com o Git, permitindo gerir repositórios e controlar a versão diretamente na IDE. Além disso, possui uma vasta biblioteca de extensões, um editor de código avançado com recursos vastos, e um terminal integrado, tudo isso contribui para aumentar a produtividade e desenvolvimento.

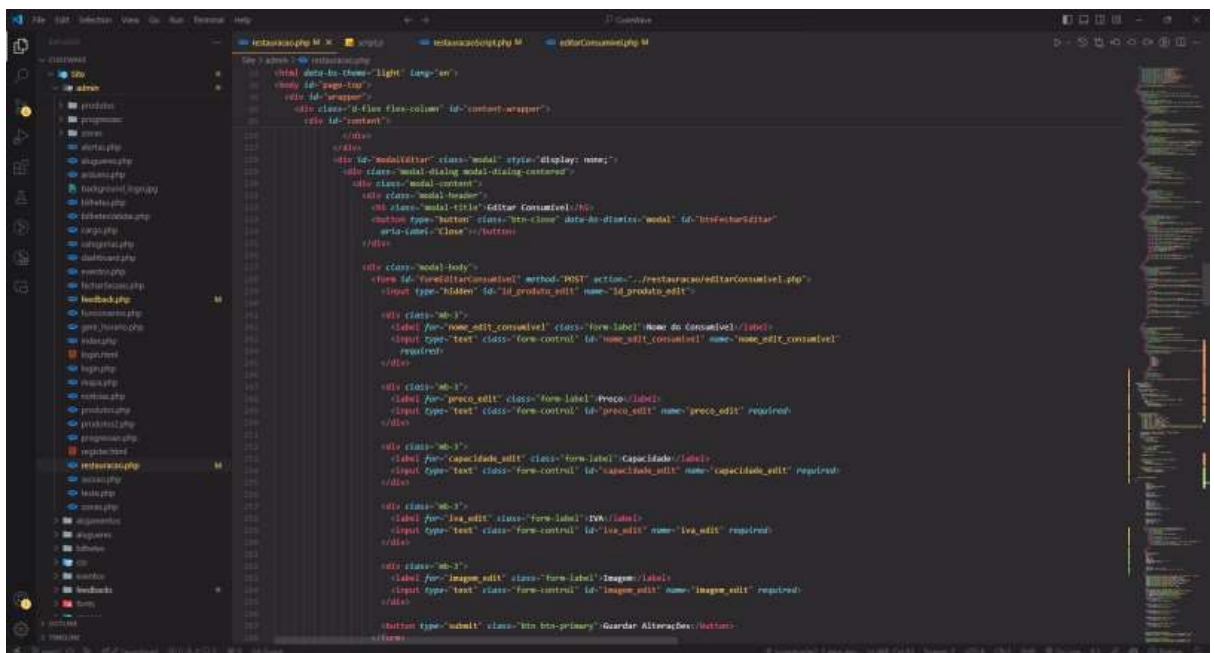


FIGURA 23 - VISUAL STUDIO CODE

- ♦ **Discord:** É uma plataforma de comunicação gratuita que permite a comunicação entre pessoas por meio de texto, voz e vídeo. Esta aplicação permite ainda a partilha de ficheiros e a criação de servidores personalizáveis. Por ser muito acessível e utilizada diariamente pelo grupo foi escolhida para organizar e resolver problemas sobre o projeto.



FIGURA 24 - DISCORD

- ♦ **Zoom:** Plataforma de comunicação por vídeo que proporciona serviços de videoconferência, reuniões online, colaboração em grupo, etc. Esta foi muito útil para as reuniões semanais que tínhamos com o professor orientador Rui Isidro.
- ♦ **Teams:** É uma plataforma de colaboração e comunicação desenvolvida pela Microsoft, que integra chat, videoconferências, armazenamento de ficheiros e integração com aplicações. Foi usada como uma das formas de comunicação com o professor Rui Isidro.

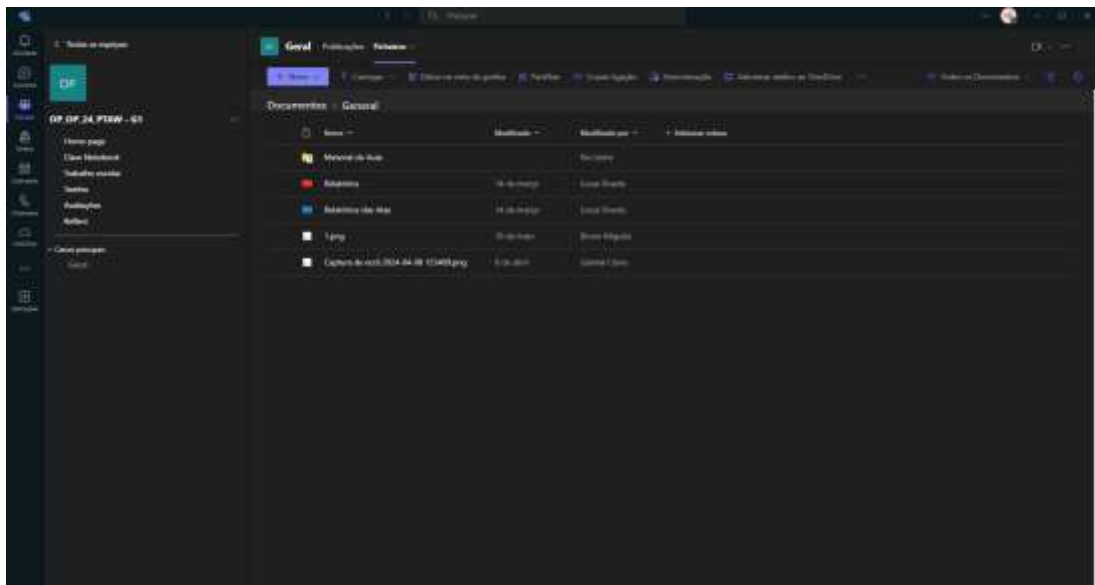


FIGURA 25 - TEAMS

- **Trello:** É uma ferramenta de gestão de projetos que utiliza um sistema de cartões e listas para ajudar a organizar tarefas de forma eficiente. É visualmente intuitivo e flexível, permitindo que os utilizadores acompanhem o progresso das suas atividades em um formato de quadro de tarefas.

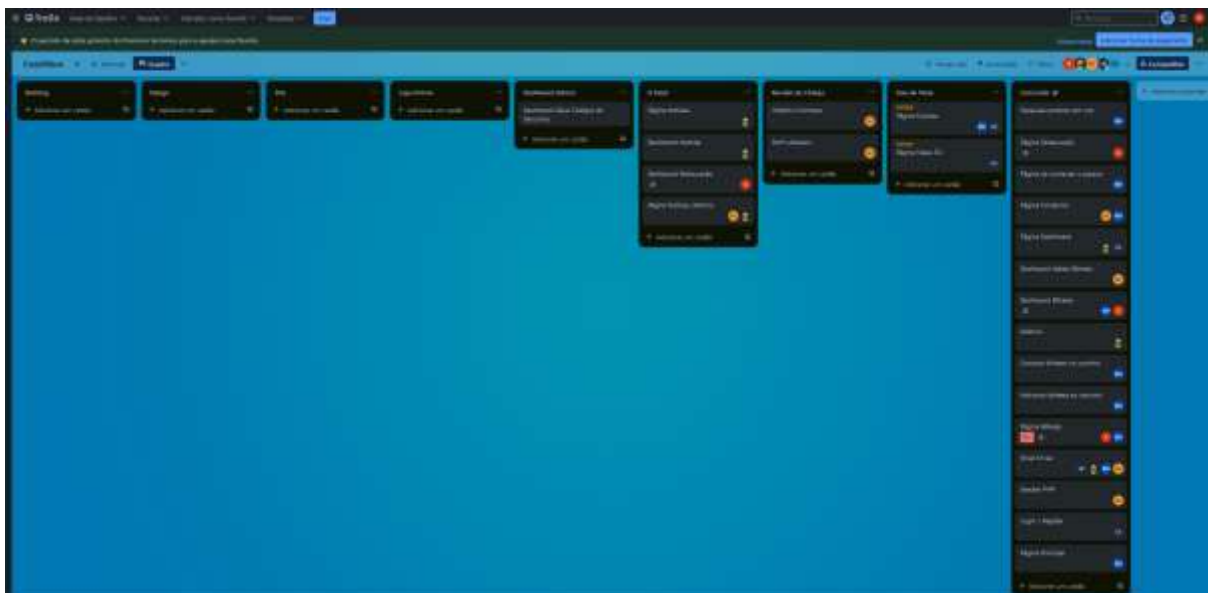


FIGURA 26 - TRELLO

- **Balsamiq:** O Balsamiq é uma ferramenta de *wireframing* que permite a criação rápida e fácil de esboços de interfaces de utilizador (UI). É especialmente útil para designers, desenvolvedores e equipas de produto que precisam de visualizar e planear a estrutura e funcionalidade de aplicações e websites antes de entrar na fase de desenvolvimento.



FIGURA 27 - BALSAMIQ

- **Figma:** O Figma é uma ferramenta de design colaborativa baseada na nuvem, utilizada para criar interfaces de utilizador (UI), protótipos e experiências de utilizador (UX). É amplamente usada por designers e equipas de desenvolvimento devido às suas poderosas funcionalidades de design, prototipagem e colaboração em tempo real.

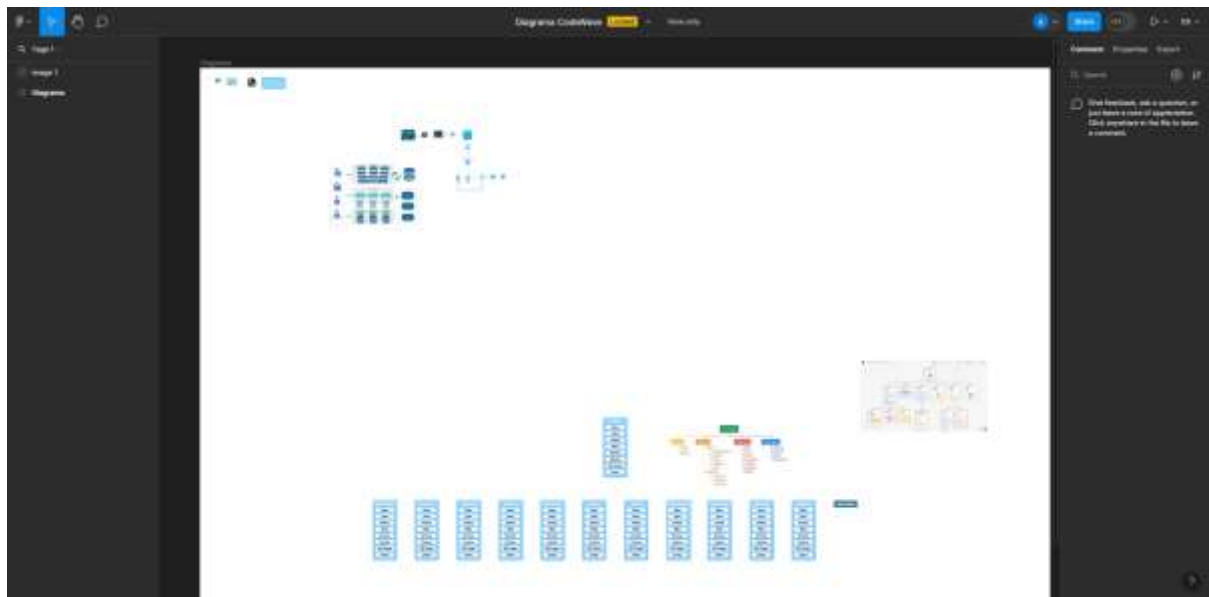


FIGURA 28 - FIGMA

10.1 Bibliotecas Externas

A aplicação web foi desenvolvida com base em um template disponibilizado pelo ThemeForest. Esse template, veio com o Bootstrap 3 pré-instalado e com jQuery integrado por padrão, serviu como uma base essencial do projeto, a partir da qual iniciamos o desenvolvimento do projeto.

Abaixo, disponibilizamos uma tabela que lista as bibliotecas de terceiros utilizadas neste projeto.

Nome	Descrição
Chart.js	Biblioteca para criação de gráficos dinâmicos e interativos
PHP Mailer	Envio programático de e-mails utilizando PHP, incluindo recursos avançados como anexos de arquivos e autenticação SMTP
TCPDF	Biblioteca utilizada para criar documentos PDF's personalizados com controlo total sobre layout e conteúdo
PHP QR Code	Criação de códigos QR em formato de imagem, amplamente utilizada para diversas aplicações, incluindo identificação e partilha de informações

TABELA 52 BIBLIOTECAS EXTERNAS

- O **Chart.js** foi usado para criar a interface principal do admin, que exibe gráficos com informações do parque e da loja online, como o número total de bilhetes vendidos, os produtos mais populares, o total de compras, os lucros e as receitas por categorias entre outros. Esses gráficos ajudam o gerente a ter uma visão rápida e clara do desempenho do parque aquático.
- O **PHP Mailer** foi utilizado para enviar e-mails tanto para o parque aquático quanto para os utilizadores, abrangendo diferentes finalidades. Foram enviadas mensagens ao parque aquático como meio de comunicação entre os utilizadores e o estabelecimento. Além disso, e-mails foram enviados aos utilizadores da loja online com detalhes sobre as suas compras de bilhetes e produtos, bem como informações sobre eventos e notícias relevantes.
- O **TCPDF** foi usado para criar as faturas da loja online e produzir relatórios a partir do painel de administração. Por meio desta solução, conseguimos criar documentos PDF das faturas para os utilizadores e relatórios detalhados com as informações da administração.
- O **PHP QR Code** foi utilizado para criar os códigos QR dos bilhetes do parque aquático. Quando um utilizador adquiria um bilhete, este recebia por e-mail um código QR contendo o número correspondente ao bilhete.

11. Desenvolvimento do website

11.1 Diagrama de Fluxo de Navegação do Website

O diagrama de fluxo de navegação de um site, consiste numa representação visual que ilustra as conexões e interações entre as diversas páginas e funcionalidades de um website. Este diagrama detalha como as diferentes seções do site estão interligadas, permitindo uma compreensão clara da estrutura de navegação.

É fundamental para identificar o percurso que os utilizadores podem seguir, facilitando a análise da usabilidade e a identificação de áreas que podem ser otimizadas para melhorar a experiência do utilizador.

11.1.1 Diagrama de Fluxo de Navegação do Website (Gerente)

O diagrama é dividido em várias seções principais, cada uma representando um conjunto de páginas e funcionalidades relacionadas:

- **Index:** Esta é a interface que contém hiperligações para as diversas outras seções de administração, incluindo Mapa do Parque, Contactos, Horários, Bilhetes, Restauração, Alojamento, Alugueres, Eventos, Notícias, Loja-Online e Feedback etc.
- **Alertas:** Esta página mostra todos os alertas, quando algum gerente adiciona alguma coisa na base de dados.
- **Categorias:** Permite ao gerente adicionar, editar ou apagar uma categoria.
- **Funcionários:** A dashboard funcionários permite ao gerente, adicionar um novo trabalhador, e incluir as suas funções no parque, como o seu cargo.
- **Cargos:** A tabela cargo, permite ao gerente criar e gerir os cargos.
- **Progressão:** Na tabela progressão o gerente pode adicionar informações extras, como salário e a data em que o funcionário entrou ao serviço.
- **Zonas:** Permite ao gerente adicionar, editar ou apagar uma zona.
- **Produtos:** Permite ao gerente adicionar, editar ou apagar uma zona, diferente das outras dashboards esta permite associar uma imagem a um produto.
- **Restauração:** Inclui a gestão e visualização de opções de como adicionar, apagar, editar e listar consumíveis, entre outras.

- **Bilhetes:** Inclui a página de listar os Bilhetes, onde os bilhetes disponíveis podem ser visualizados e validados.
- **Mapa Interativo:** Contém a página de um mapa interativo do parque, onde é possível controlar as luzes do parque, ver as zonas no mapa e os eventos que estão a decorrer.
- **Horários:** Visualização dos horários e atualização dos mesmos.
- **Feedback:** Abrange as páginas para mostrar e gerir os *feedbacks* dos utilizadores.
- **Eventos:** Mostra os eventos, e permite ao gerente entrar em contacto com o utilizador que pediu o evento.
- **Alojamento:** Lista todos os alojamentos e permite aceitar as reservas.
- **Alugueres:** Lista os alugueres com a opção de verificar os alugueres.
- **Notícias:** Apresenta as notícias com opções de adicionar, editar e apagar as mesmas.
- **Loja-Online:** Apresenta o endpoint /loja-online para acesso à loja virtual.
- **Contactos:** Contém a página para entrar em contacto com o parque.

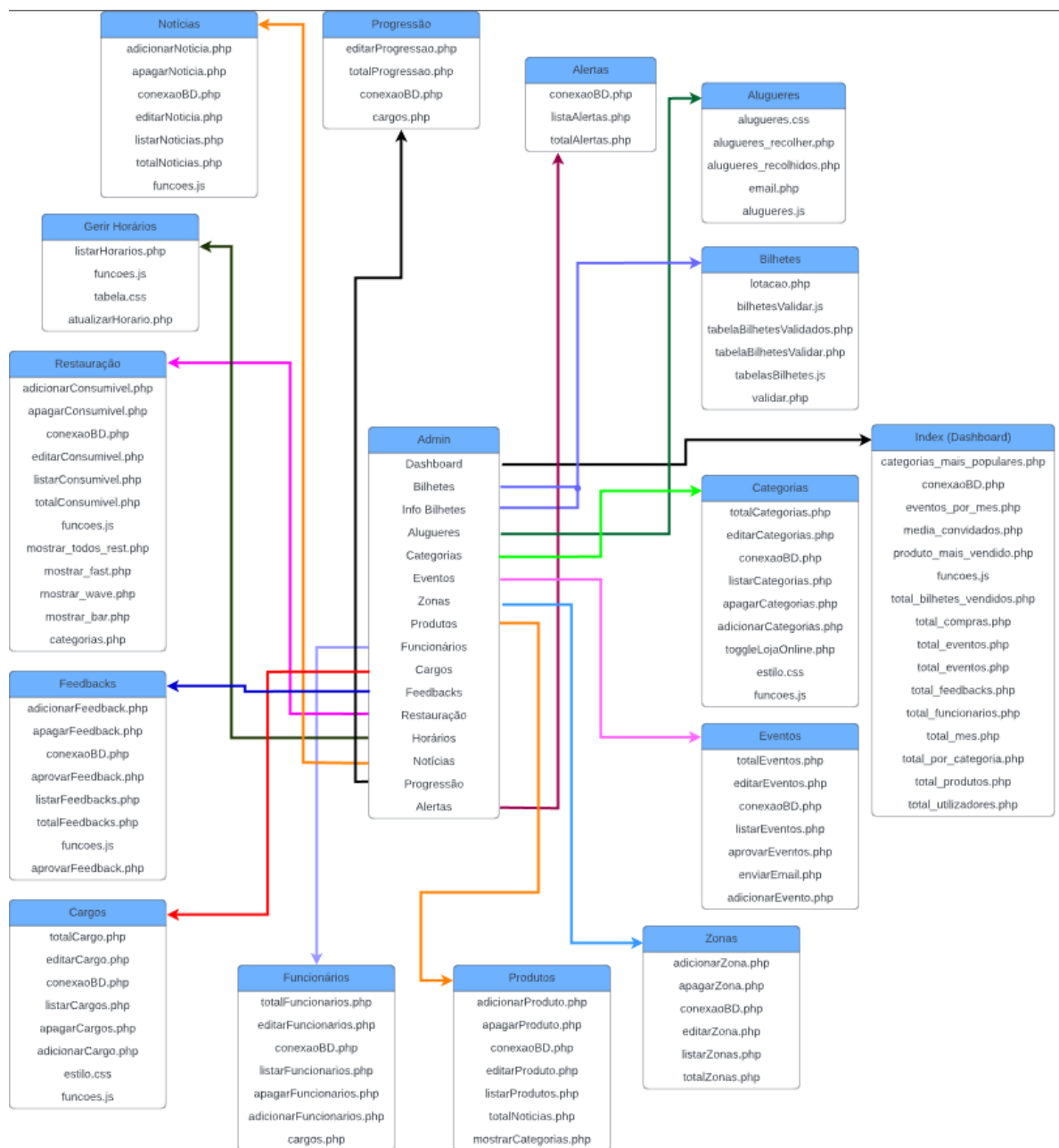


FIGURA 29 - DIAGRAMA DE FLUXO DE NAVEGAÇÃO DO WEBSITE (ADMIN)

11.1.2 Diagrama de Fluxo de Navegação do Website (Site)

- **Index:** A página do index, é a página principal do website. Contem um menu com todas as referências para as outras páginas, incluindo a da loja-online.
- **Alojamento:** Página onde o utilizador pode escolher um tipo de bungalow para passar uns dias no parque.
- **Bilhetes:** Página de compra de bilhetes para visitar o parque aquático.
- **Mapa Interativo:** Contém a página de um mapa interativo do parque, onde é possível controlar as luzes do parque, ver as zonas no mapa e os eventos que estão a decorrer.
- **Horários:** Visualização dos horários do parque aquático.
- **Feedback:** Permite aos utilizadores ver os *feedbacks* de outros visitantes e escrever uma opinião também.
- **Eventos:** Mostra os eventos, organizados por meses que aconteceram no parque.
- **Alugueres:** Os utilizadores podem alugar equipamentos quando forem visitar o parque.
- **Notícias:** Apresenta as notícias mais recentes do parque.
- **Loja-Online:** Apresenta o endpoint /loja-online para acesso à loja virtual.
- **Contactos:** Contém a página para entrar em contacto com o parque.
- **Restauração:** Mostra os menus, com os preços dos diferentes restaurantes e do bar.

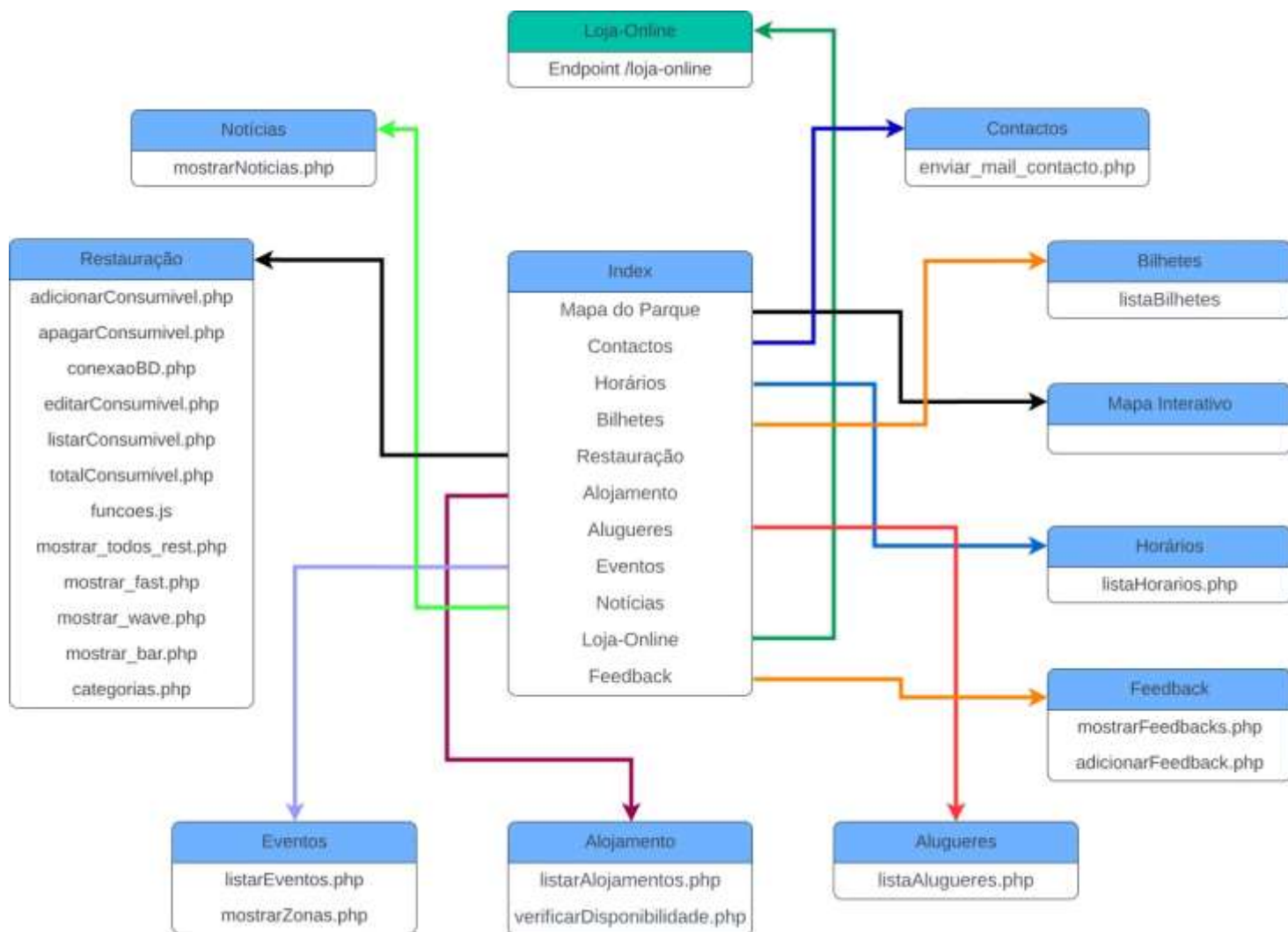


FIGURA 30 - DIAGRAMA DE FLUXO DE NAVEGAÇÃO DO WEBSITE (SITE)

11.1.3 Diagrama de Fluxo de Navegação do Website (Loja-Online)

- **Index:** A página principal da loja-online mostra os produtos agrupados por categoria, incluindo uma secção com os produtos mais recentes.
- **Categoria:** Inclui os produtos da loja, agrupados por categorias específicas.
- **Produtos:** Cada produto contém uma imagem, título e preço.
- **Perfil:** O perfil de utilizador da loja permite aos utilizadores verem os seus bilhetes comprados anteriormente e o seu histórico de compras.
- **Entrega:** Página com a lógica de envio da compra para a morada do utilizador.
- **Pagamento:** Oferece 3 métodos de pagamento, que o utilizador pode escolher.
- **Confirmação:** Mostra um resumo da compra do utilizador, para confirmar se todos os detalhes estão bem preenchidos.
- **Encomenda Finalizada:** No fim de confirmar a compra, o utilizador é redirecionado para uma página a dizer que a sua compra foi efetuada com sucesso e é redirecionado novamente para a página principal da loja-online.

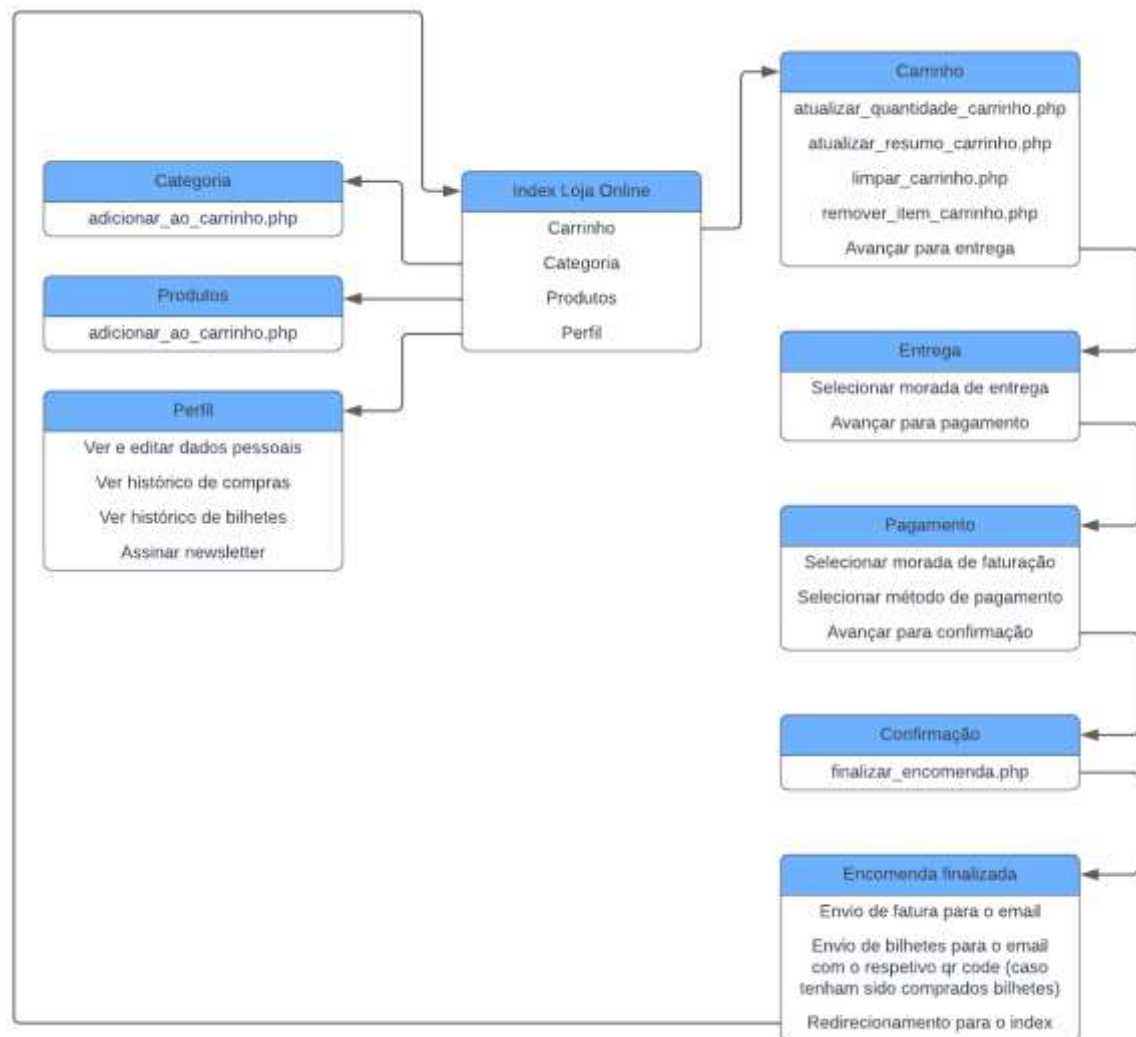


FIGURA 31 - DIAGRAMA DE FLUXO DE NAVEGAÇÃO DO WEBSITE (LOJA-ONLINE)

11.2 Modelo de Dados Persistente

A necessidade de guardar dados de modo persistente fomentou a criação de um modelo de base de dados, seguido da sua implementação.

A base de dados modelada, como se observa nas figuras abaixo, foi criada através da interpretação da lista de requisitos. O modelo foi criado recorrendo à aplicação de modelação Visual Paradigm. Após a aprovação do modelo criado, que representaria as necessidades resultantes dos requisitos, implementou-se o Sistema de Gestão de Bases de Dados num servidor remoto. Na Base de dados ainda foi guardada uma *procedure* para a criação da mesma com o script que está no **ANEXO D**.

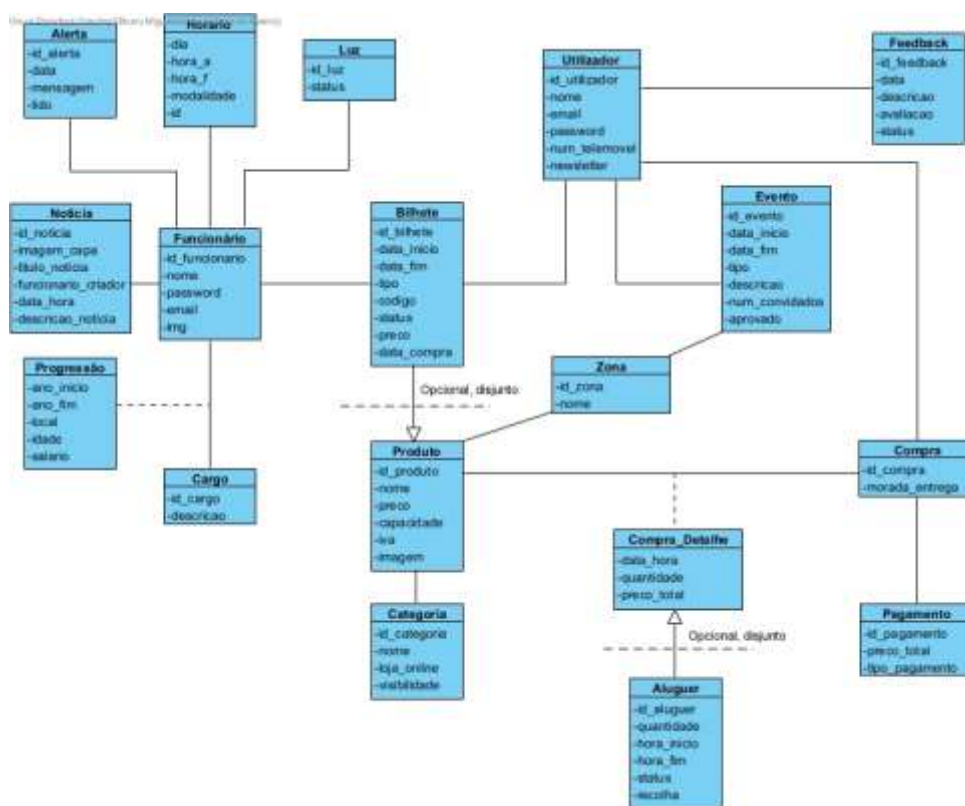


FIGURA 32 - MODELO DE BASE DE DADOS CONCEPTUAL

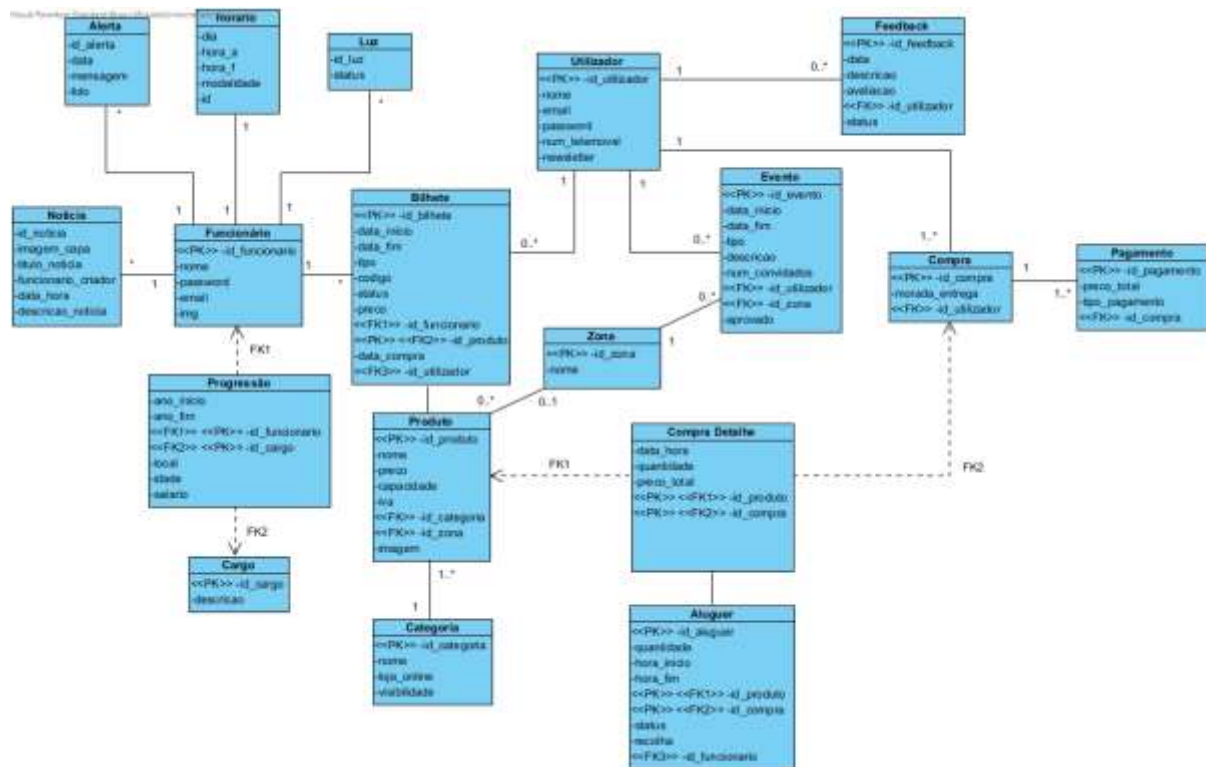


FIGURA 33 - MODELO DE BASE DE DADOS RELACIONAL

12. Interface

Uma interface de utilizador é o ponto de interação entre humanos e máquinas. O seu objetivo é permitir uma operação entre máquina e utilizador, fornecendo *feedback* que auxilia na tomada de decisões operacionais. A interface inclui componentes de hardware (físicos) e software (lógicos), oferecendo meios de entrada para o utilizador manipular o sistema e meios de saída para que o sistema apresente as respostas às ações do utilizador.

12.1 Página principal

A página principal é a porta de entrada para a experiência completa no website do CodeWave. Nela, há um pop-up de venda de bilhetes que acompanha o cliente durante o *scroll*. A página apresenta um breve "Sobre Nós" para que o cliente conheça o CodeWave, seguido de uma galeria de fotos do parque, organizada por categorias. Após a galeria, é possível ver o mapa do parque, que destaca todas as suas funcionalidades, como fotografias em diversos pontos e imagens em 360°. Há também uma seção de *feedbacks* de clientes que visitaram o CodeWave e uma lista de eventos próximos. Além disso, a página inclui a previsão meteorológica para os próximos 7 dias.

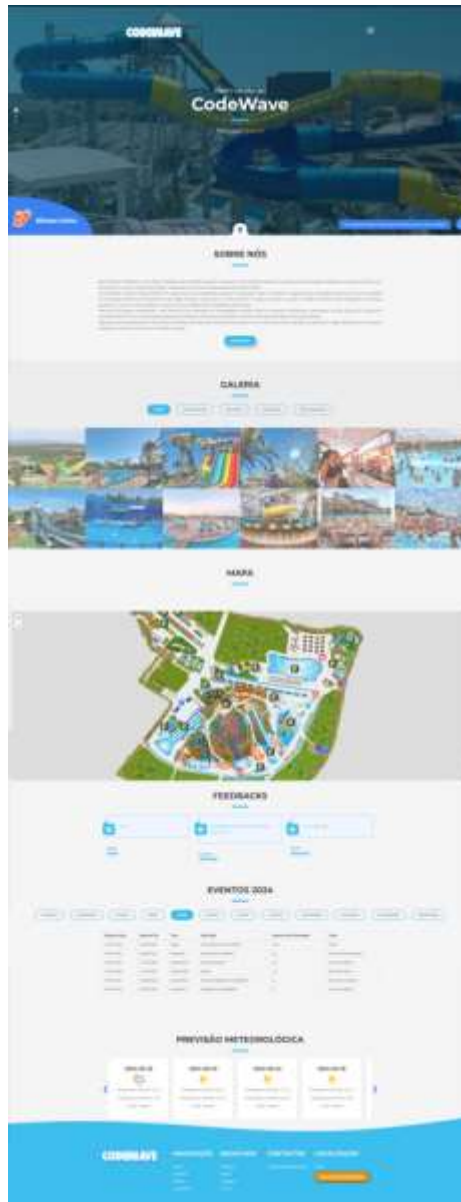


FIGURA 34 - PÁGINA PRINCIPAL

12.2 Página do mapa interativo

Na página do mapa interativo é possível que o utilizador conheça o nosso parque aquático! Com uma planta realista, onde é possível identificar os locais de relva, as diversões aquáticas, os WC e até mesmo a loja de lembranças do CodeWave. Os pins têm uma foto do local assim como uma breve descrição. Onde tem uma camara fotográfica é possível observar o local em 360°.

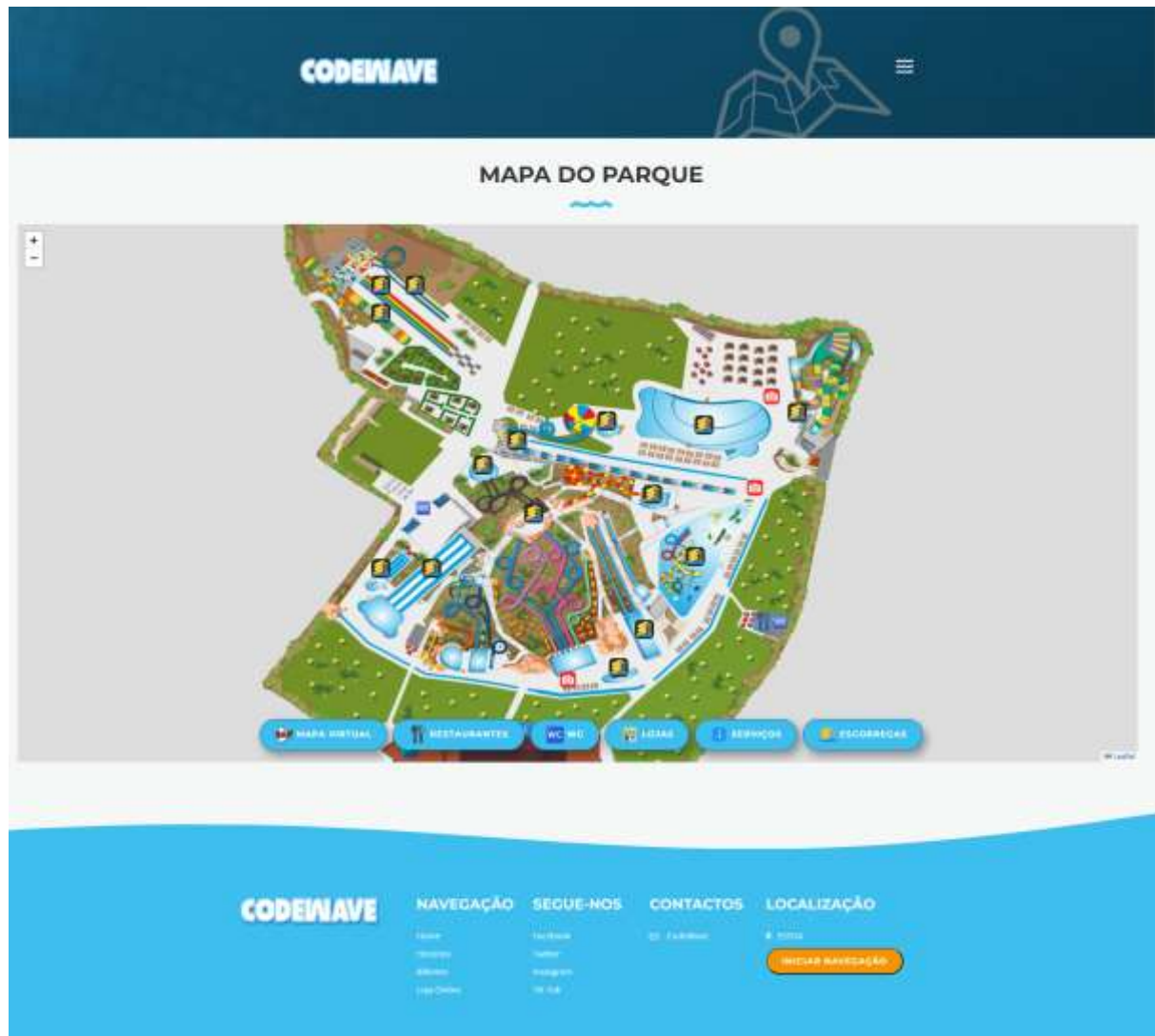


FIGURA 35 - PÁGINA DO MAPA INTERATIVO

12.3 Página contactos

A página de contatos do website do CodeWave é fundamental para a comunicação com os visitantes. Nela, os clientes encontram um formulário de contato que facilita o envio de mensagens diretamente à equipa do CodeWave. A página também exibe informações detalhadas, como endereço, email, há links diretos para as redes sociais do CodeWave, permitindo que os visitantes se conectem e acompanhem as atualizações em diversas plataformas.




The screenshot shows the 'CONTACTOS' (Contacts) page of the CodeWave website. The page has a dark blue header with the CodeWave logo and icons for a phone, location pin, email, and social media. Below the header, the main content area is light gray and features the heading 'ENTRA EM CONTACTO CONNOSCO!' (Get in touch with us!). Under this heading is a contact form with four input fields: 'Nome completo' (Full name), 'Número de telemóvel' (Mobile number), 'Assunto' (Subject), and 'Escreva a sua mensagem...' (Write your message...). A blue 'SUBMITER' button is located below the form. The footer is a solid blue bar containing the CodeWave logo, navigation links (Home, Horários, Serviços, Loja Online), social media links (Facebook, Twitter, Instagram, TikTok), contact information (E2, CodeWave, ESTGA), and a location pin icon with the text 'INICIAR NAVEGAÇÃO' (Start navigation).

FIGURA 36 - PÁGINA CONTACTOS

12.4 Página horários

A página "Horários" do site da CodeWave fornece informações sobre o horário de funcionamento do parque. Apresenta uma tabela bem estruturada que exibe os horários de abertura e fecho para cada dia da semana, com uma nota clara indicando que o parque está fechado às quartas-feiras. A simplicidade visual da página facilita aos visitantes dando a compreensão rápida das informações essenciais de que precisam para planear sua visita.


CODENAVE

Geral		Escolas:
Data	Hora de Abertura	Hora de Fecho
Domingo	09:00	20:00
Segunda-feira	10:00	18:00
Terça-feira	10:00	18:00
Quarta-feira	Encerrado	Encerrado
Quinta-feira	14:00	20:00
Sexta-feira	10:00	18:00
Sábado	09:00	20:00

CODENAVE

- NAVEGAÇÃO
- SEGUIE-NOS
- CONTACTOS
- LOCALIZAÇÃO

[Home](#)
[Reservas](#)
[Atividades](#)
[Jogos Educativos](#)

[Parceiros](#)
[Partidos](#)
[Entrevistas](#)
[Ola, Tula!](#)

[EN](#) / [PT](#) / [FR](#)

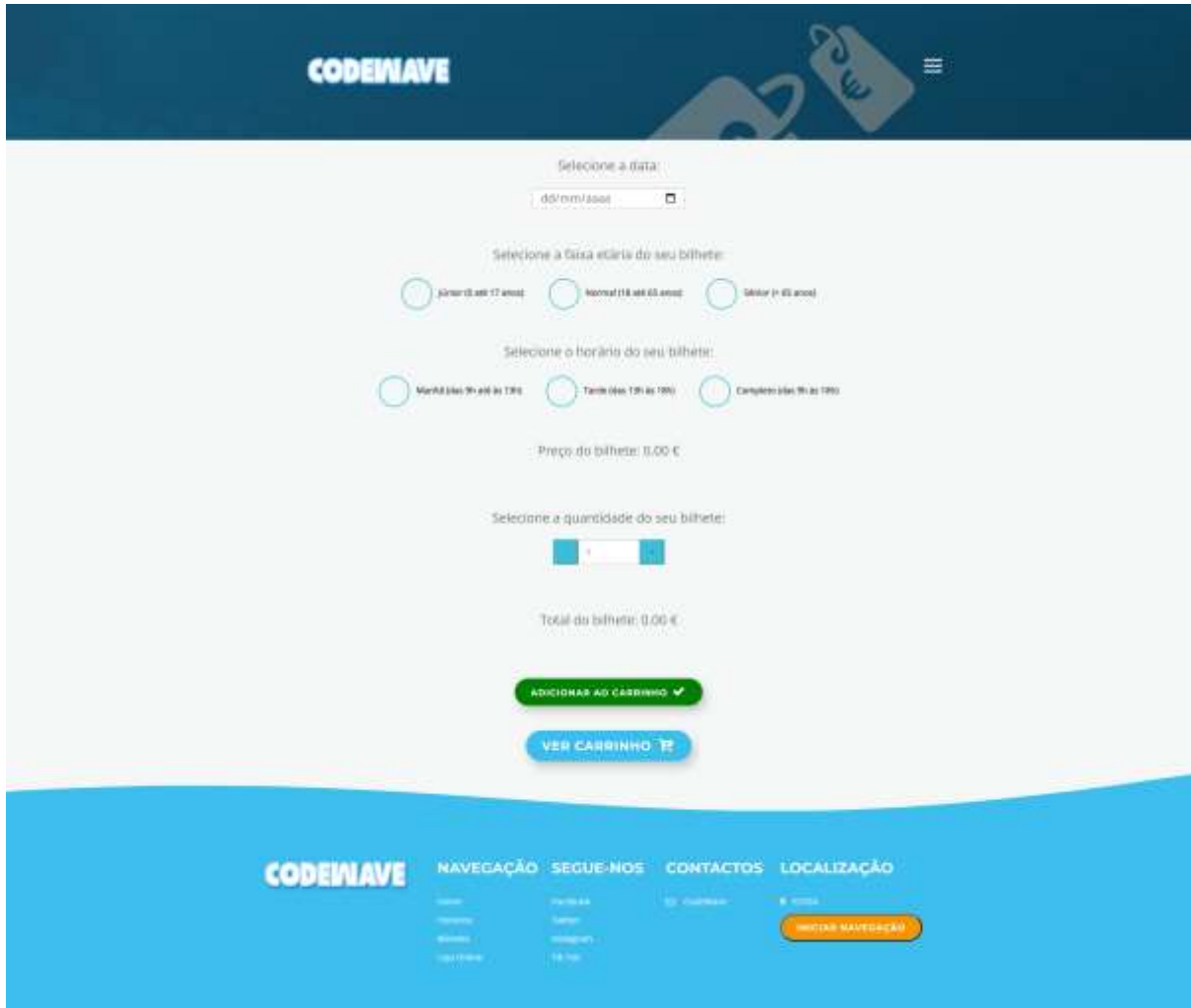
+ 0704
INICIAR NAVEGAÇÃO

Não se esqueça de voltar a ligar aqui para a "Nova Edição e Actualização"

FIGURA 37 - PÁGINA HORÁRIOS

12.5 Página bilhetes

Na página de bilhetes do CodeWave, o utilizador seleciona a data da visita, a faixa etária do bilhete (júnior, normal, sénior), o horário (manhã, tarde, completo) e a quantidade desejada. O total é calculado automaticamente e um botão "Adicionar ao Carrinho" e “Ver carrinho” permite finalizar a compra de forma prática. Esta página foi projetada para oferecer uma experiência de compra rápida e fácil.



CODENAVE

Selecione a data:

dd/mm/aaaa

Selecione a faixa etária do seu bilhete:

☐ Júnior (3 até 12 anos) ☐ Normal (13 até 65 anos) ☐ Sénior (+ 65 anos)

Selecione o horário do seu bilhete:

☐ Manhã (das 9h até às 13h) ☐ Tarde (das 13h às 18h) ☐ Noite (das 19h às 23h)

Preço do bilhete: 0.00 €

Selecione a quantidade do seu bilhete:

1

Total do bilhete: 0.00 €

ADICIONAR AO CARRINHO ✓

VER CARRINHO 🛒

CODENAVE

NAVEGAÇÃO | SEGUIE-NOS | CONTACTOS | LOCALIZAÇÃO

Home | Quem somos | Onde estamos | Mapa

Restaurante | Bar | Loja | Loja online

INICIAR NAVEGAÇÃO

FIGURA 38 - PÁGINA BILHETES

12.6 Página restauração

Os visitantes do CodeWave encontram na seção de restauração três botões principais: Restaurante Wave, Restaurante Fast e Bar. Cada botão revela menus específicos, incluindo imagens, nomes e preços dos consumíveis. Além disso, um botão "Mostrar tudo" está disponível, permitindo visualizar todos os consumíveis dos restaurantes em conjunto. Esta organização facilita a escolha do local para comer, apresentando todas as informações de forma clara e atrativa.

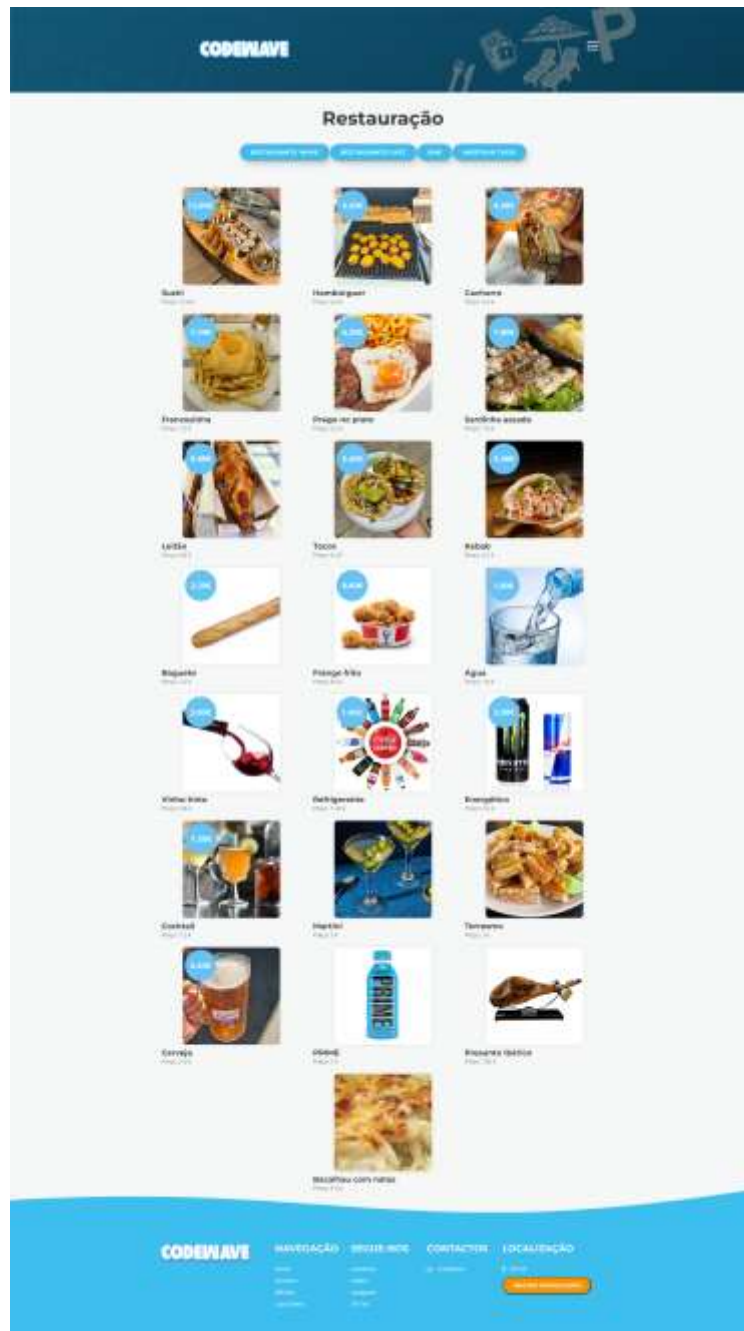


FIGURA 39 - PÁGINA RESTAURAÇÃO

12.7 Página alojamento

Na página de alojamento do CodeWave, os utilizadores podem explorar três opções de bungalows: Basic, XL e Premium. Cada tipo de alojamento está claramente apresentado com uma imagem, o nome e o preço por dia. Há também um botão "Reservar" abaixo de cada opção, permitindo que os visitantes façam suas reservas de forma rápida e conveniente. Esta organização facilita a comparação das diferentes opções de alojamento, ajudando os utilizadores a escolher a melhor opção para a sua estadia no parque.

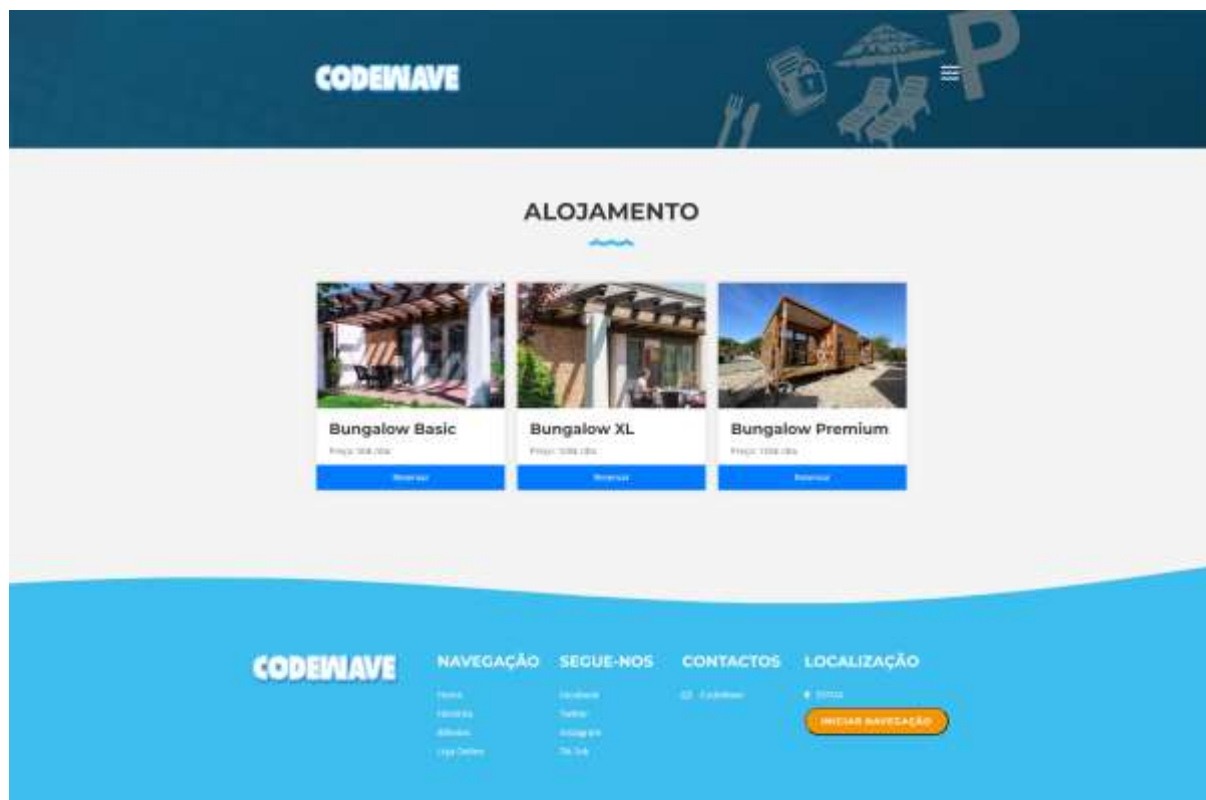


FIGURA 40 - PÁGINA ALOJAMENTO

12.8 Página alugueres

Na página de alugueres do CodeWave, os utilizadores podem alugar espreguiçadeiras e guarda-sóis. Cada produto é exibido com uma imagem, o nome e o preço por hora. Abaixo de cada opção, um botão "Alugar" permite que os visitantes façam a reserva do produto desejado de forma rápida e fácil. Esta disposição ajuda os utilizadores a escolher e alugar os equipamentos necessários para desfrutar ao máximo do parque aquático.

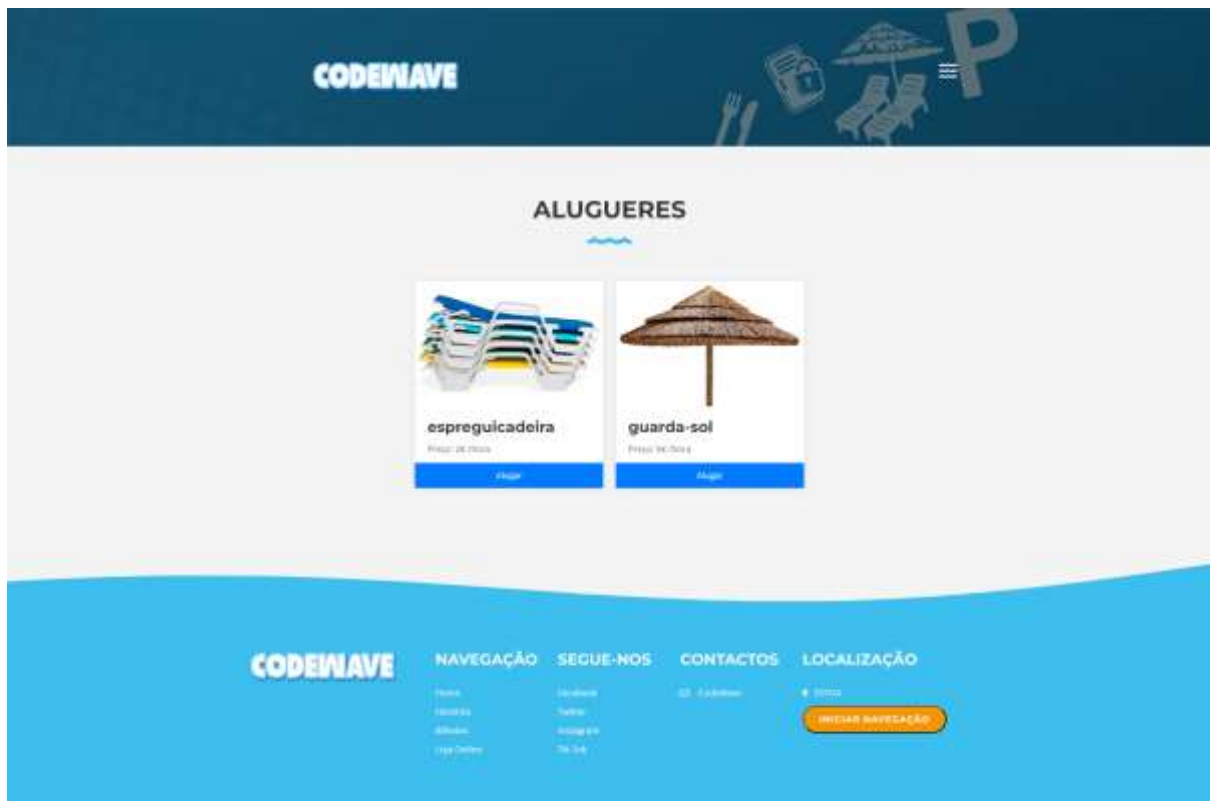


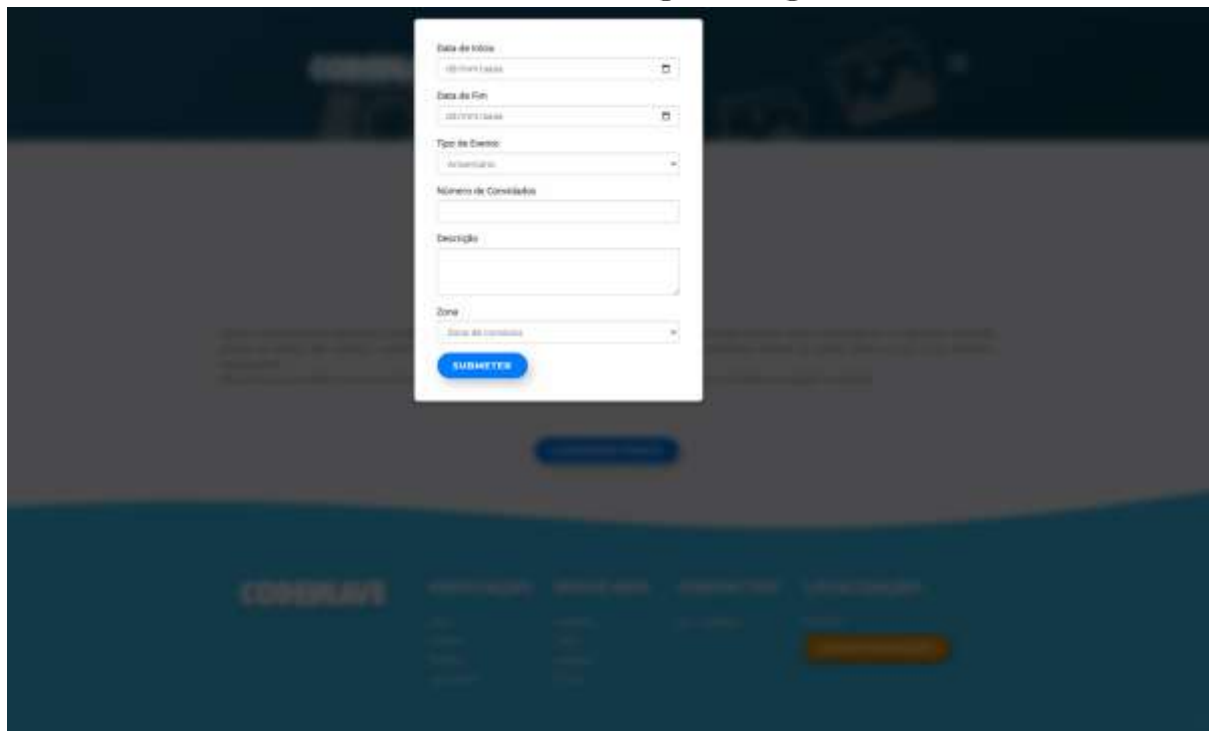
FIGURA 41 - PÁGINA ALUGUERES

12.9 Página eventos

A página de eventos permite que utilizadores autenticados agendem eventos no parque. Após o login, os utilizadores podem clicar no botão "Agendar Evento" para abrir um formulário modal. O formulário solicita informações como data de início e fim, tipo de evento, número de convidados, descrição e zona do parque.



FIGURA 42 - PÁGINA EVENTOS



Data de Inicio
02/04/2024

Data do Fim
02/07/2024

Tipo de Evento
Autocurativo

Número de Consultas
1

Descrição
Evento de formação para técnicos de tratamento de águas residuais.

Zona
Zona de tratamento de águas residuais

SUBMITER

FIGURA 43 - PÁGINA CRIAR EVENTO

12.10 Página notícias

As notícias do CodeWave mantêm os utilizadores informados sobre as últimas novidades e eventos do parque. Com uma interface clara, os visitantes podem aceder a atualizações recentes, anúncios especiais e informações sobre próximos eventos. O design da página é limpo e organizado, facilitando a navegação e a leitura das notícias publicadas. Ao clicar no botão “Ficar a par das notícias mais recentes!”, se o utilizador tiver uma conta, vai ficar inscrito para receber notificações sobre as notícias mais recentes.

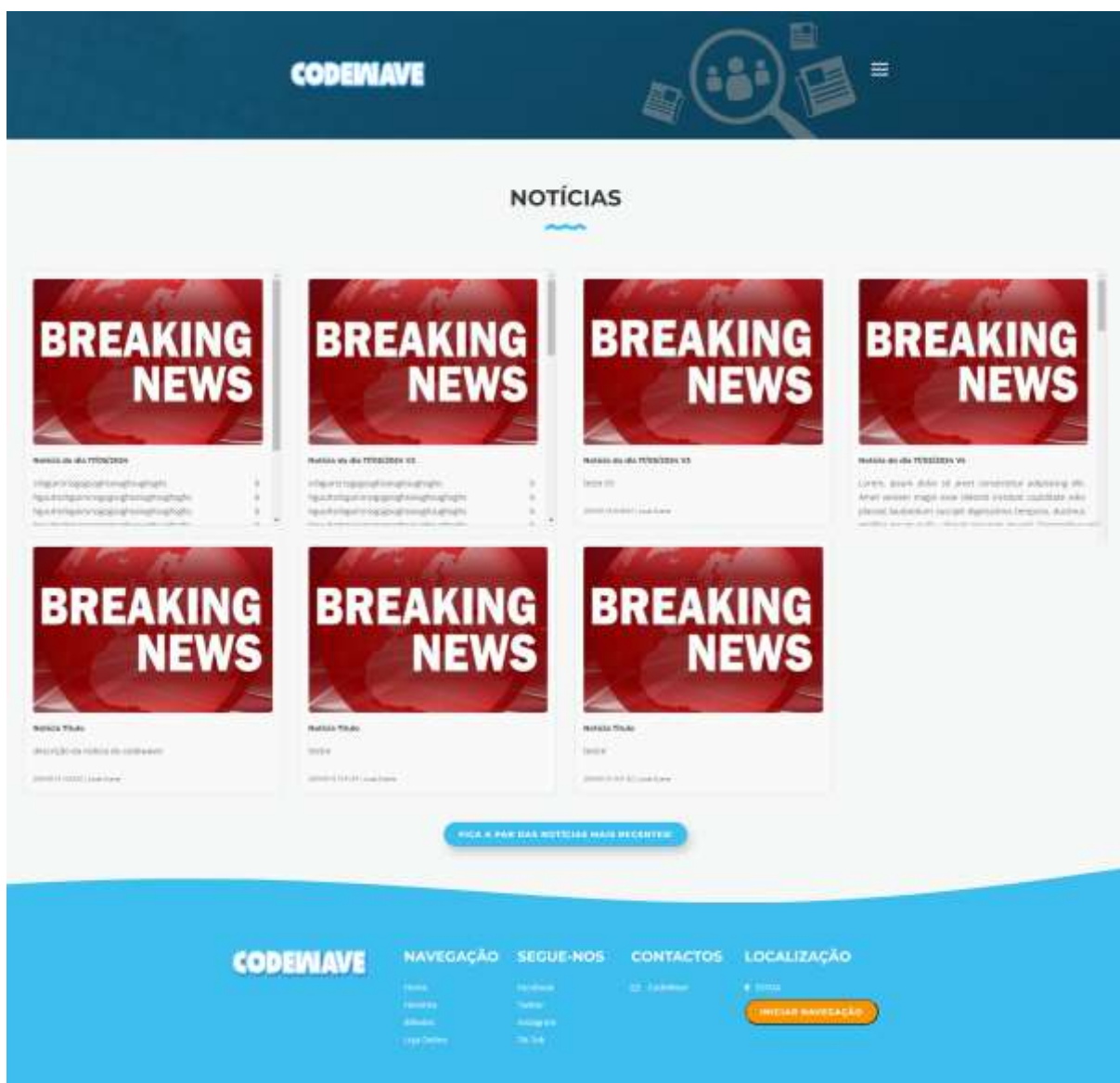


FIGURA 44 - PÁGINA EVENTOS

12.11 Página loja online

A página da loja online permite ao utilizador realizar compras de produtos. Temos uma vasta variedade de produtos, desde produtos de piscina a *merchandising*.

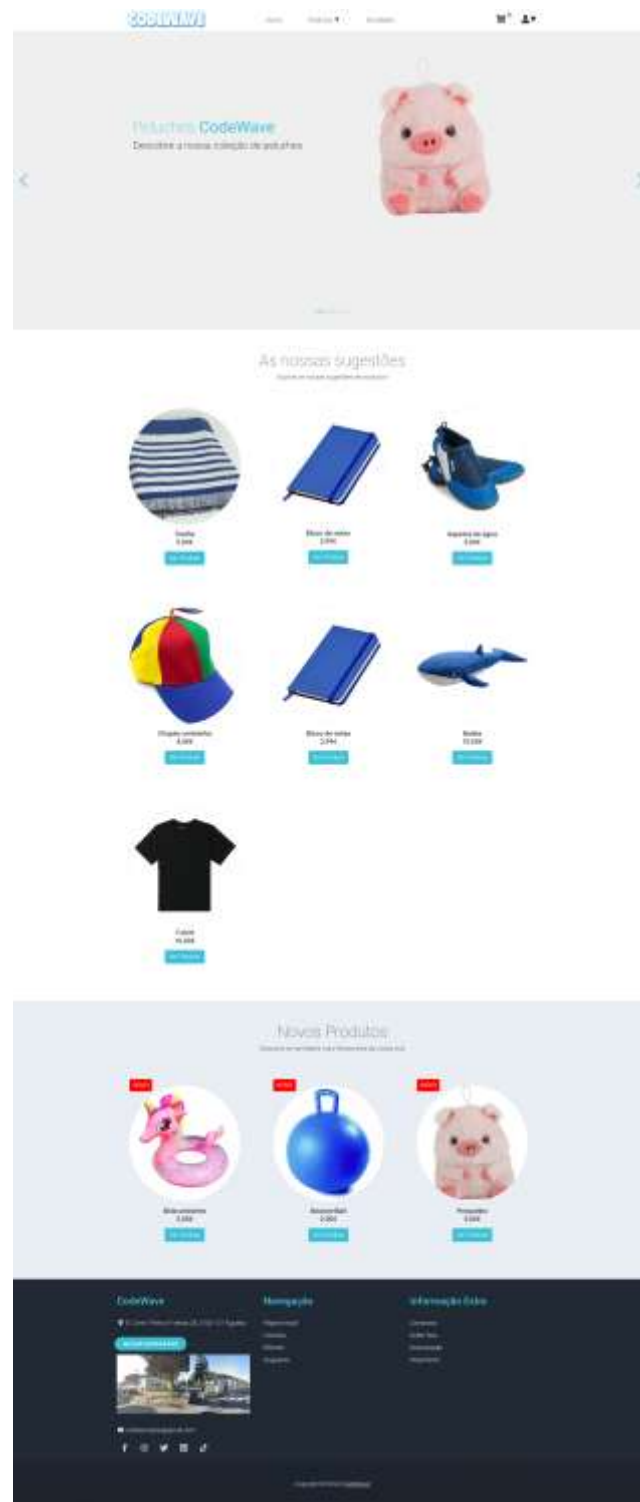


FIGURA 45 - PÁGINA LOJA ONLINE

12.12 Página loja online produtos

Aqui o cliente pode visualizar todos os produtos dispostos por categorias. Torna toda a experiência mais agradável e organizada. Esta página ainda disponibiliza a opção de procurar o produto por texto ou voz, desta forma vão aparecer os produtos que contenham os caracteres que foram referidos. Esta funcionalidade oferece uma maneira conveniente e eficiente de encontrar produtos, permitindo aos utilizadores uma experiência de pesquisa mais intuitiva e personalizada.

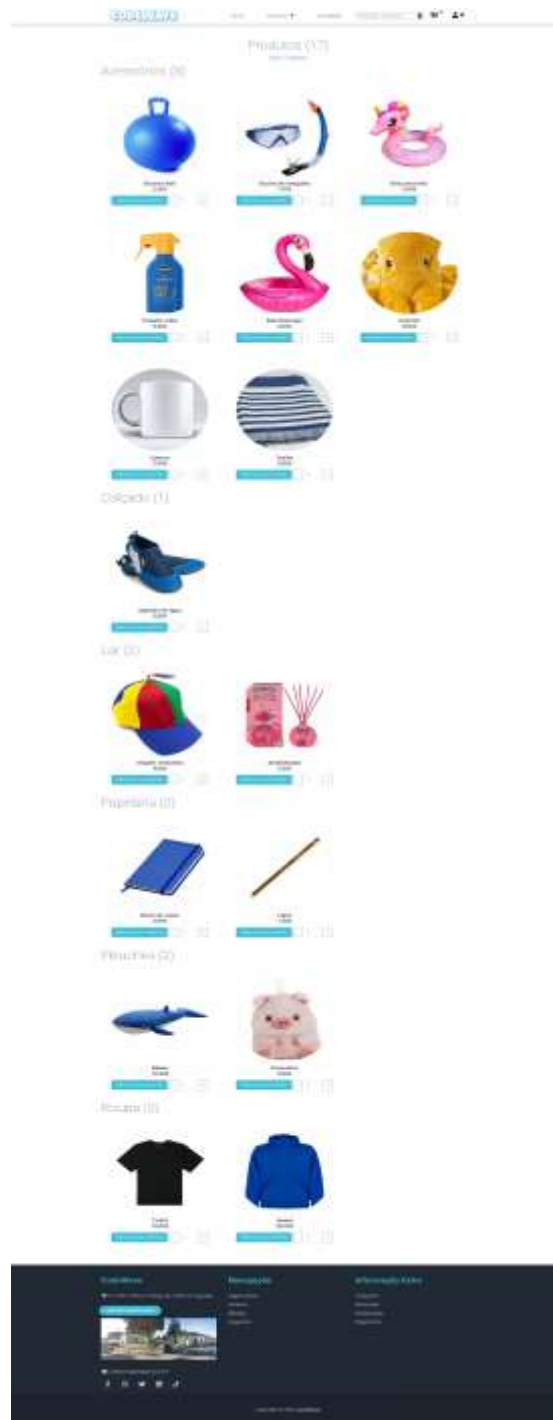


FIGURA 46 - PÁGINA LOJA ONLINE PRODUTOS

12.13 Página *login* e *signup*

A página de *login* e *signup* permite que o utilizador crie a sua conta/inicie sessão, de modo a ter acesso a imensas funcionalidades dentro do nosso website tal como deixar *reviews*, realizar compras na loja, realizar a reserva de um alojamento etc.



FIGURA 47 - PÁGINA LOGIN UTILIZADOR




FIGURA 48 - PÁGINA REGISTO UTILIZADOR

12.14 Página carrinho de compras



A página de carrinho de compras permite visualizar o resumo da compra do utilizador, permite visualizar uma foto, o preço e a quantidade do que foi selecionado, assim como o preço total. O cliente poderá alterar quantidades ou até remover os produtos se assim quiser. A compra poderá ser finalizada ou voltar a loja para escolher mais produtos.



FIGURA 49 - PÁGINA CARRINHO DE COMPRAS




[Home](#)
[Produtos](#)
[Novidades](#)

Confirmação

Por favor, antes de finalizar a sua encomenda reveja todos os detalhes e confirme se todos os dados estão corretos.

Detalhes do pedido

Imagem	Nome do Artigo	Preço	Quantidade	Subtotal
	Óculos de sol (COW-0001)	12 €	1	12,00 €
Total de Produtos				1
Subtotal				12,00 €
Custos de envio				3,00 €
IVA (23%)				2,76 €
Total				17,76 €

Dados de entrega

Nome: Art

Número de Telefone: 123456789

Opção de entrega:

Morada de Entrega

Dados de faturação

Nome: Art

Número de Telefone: 123456789

Morada de faturação: Rua Teste 123

Método de envio

Envio para o parque

Método de pagamento


Aut. WAVE

[Voltar](#)
[Finalizar encomenda](#)






CodeWave

📍 R. Gera. Pólo e Pólo 38, 3793-327 Águeda

[Ver mais detalhes](#)



📧 codewave@psd.pt

Navegação

- [Página inicial](#)
- [Produtos](#)
- [Sobre](#)
- [Águeda](#)

Informação Extra

- [Contacto](#)
- [Sobre Nós](#)
- [Reservação](#)
- [Alugamento](#)

Copyright © 2024 CodeWave

FIGURA 50 - FINALIZAÇÃO DA COMPRA

Após finalizada a compra, o cliente receberá no email da sua conta uma mensagem que contém informação sobre o bilhete.

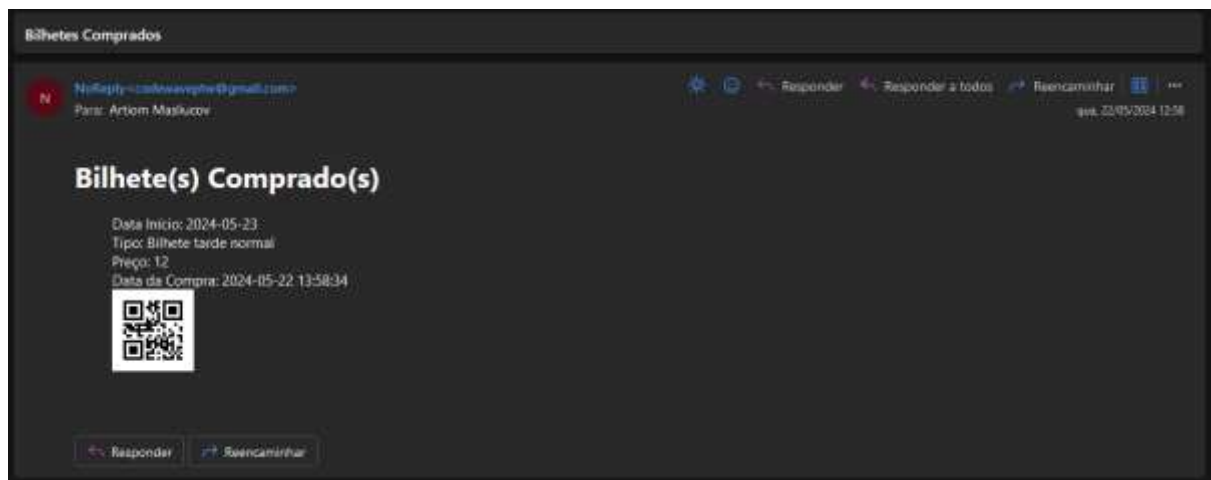


FIGURA 51 - BILHETE ELETRÓNICO

Além da informação do bilhete é também enviado para o email a fatura da compra efetuada.



Fatura 665632c5cb32a

Data: 2024-05-28

Nome do Cliente: Gabriel Cravo

Email do Cliente: gcravo@ua.pt

Morada de Entrega: R. Cmte. Pinho e Freitas 28, 3750-127 Águeda

Tipo de Pagamento: Multibanco

Detalhes dos Produtos

Produto ID	Nome do Produto	Quantidade	Subtotal
25	Baleia	2	21.00

Total: €21,00

FIGURA 52 - FATURA

12.15 Página feedbacks

Na página de *feedbacks* é possível que um utilizador do CodeWave deixe um *feedback*, quer seja uma experiência que tenha tido no parque, quer seja algo que sentiu falta no website. A página de *feedbacks* ajuda imenso a equipa a perceber as necessidades do público.

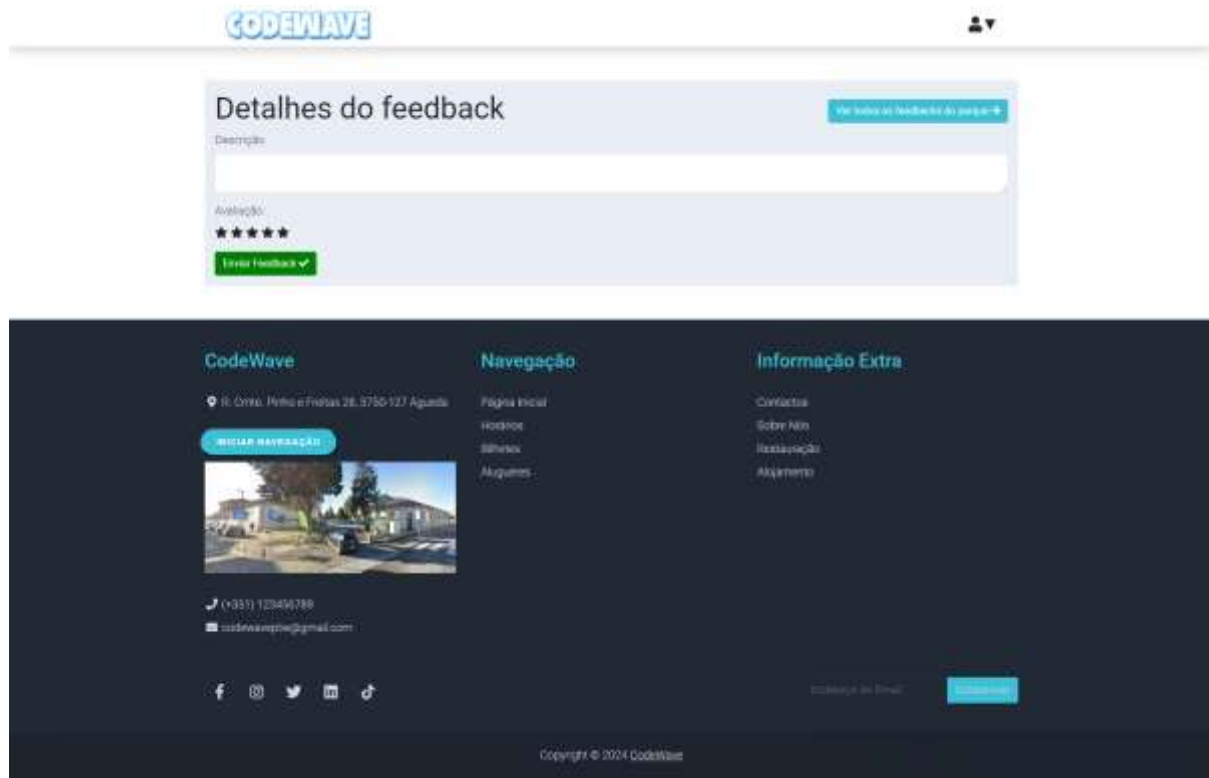


FIGURA 53 - PÁGINA FEEDBACKS

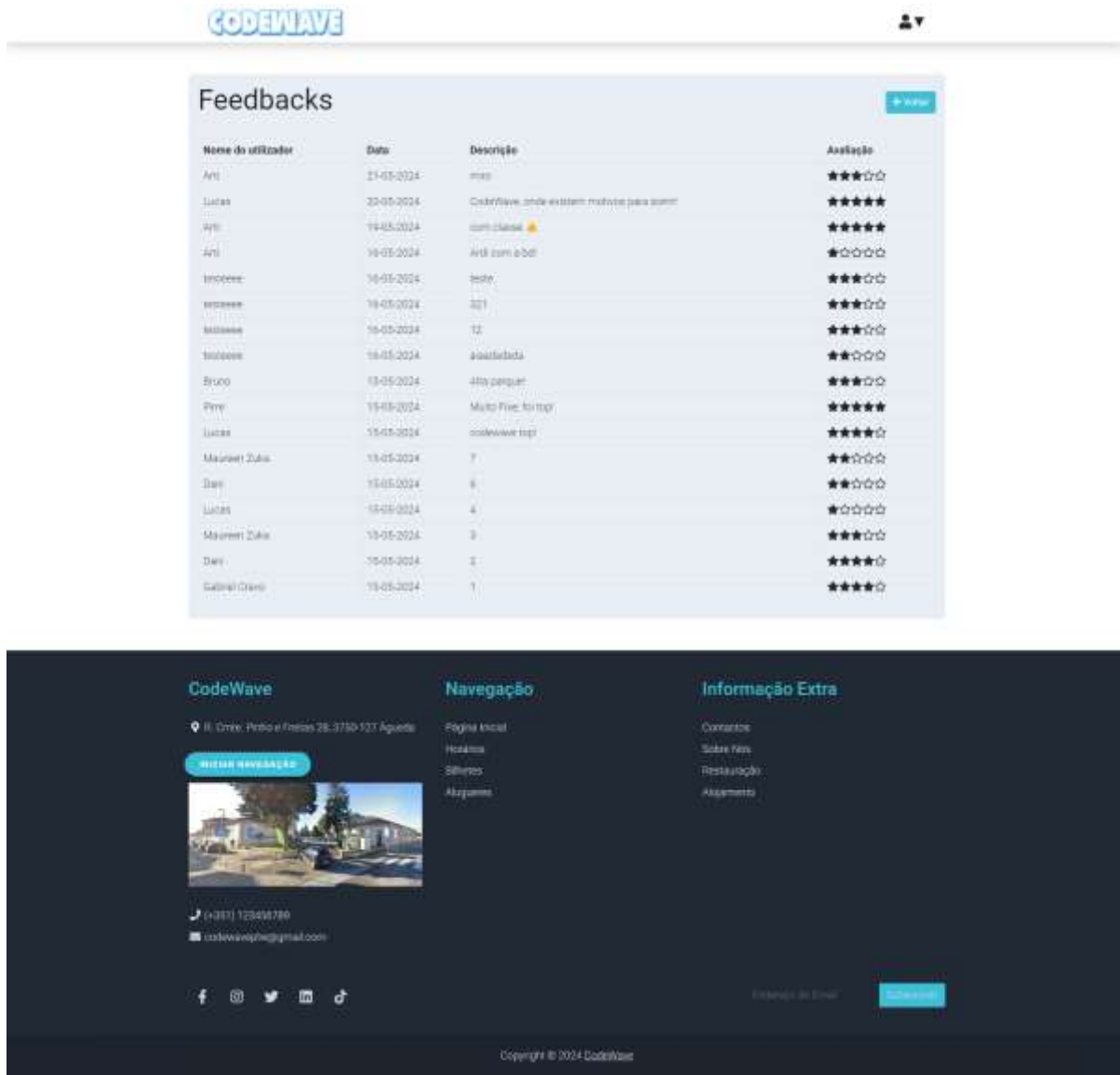


FIGURA 54 - PÁGINA COM TODOS FEEDBACKS

12.16 Dashboard admin

A dashboard de admin é disponível através de uma conta de gerente. Dá acesso a todos os dados e estatística sobre o parque. Permite também fazer diversas alterações como zonas, produtos etc, como vai ser possível observar nas imagens seguintes.

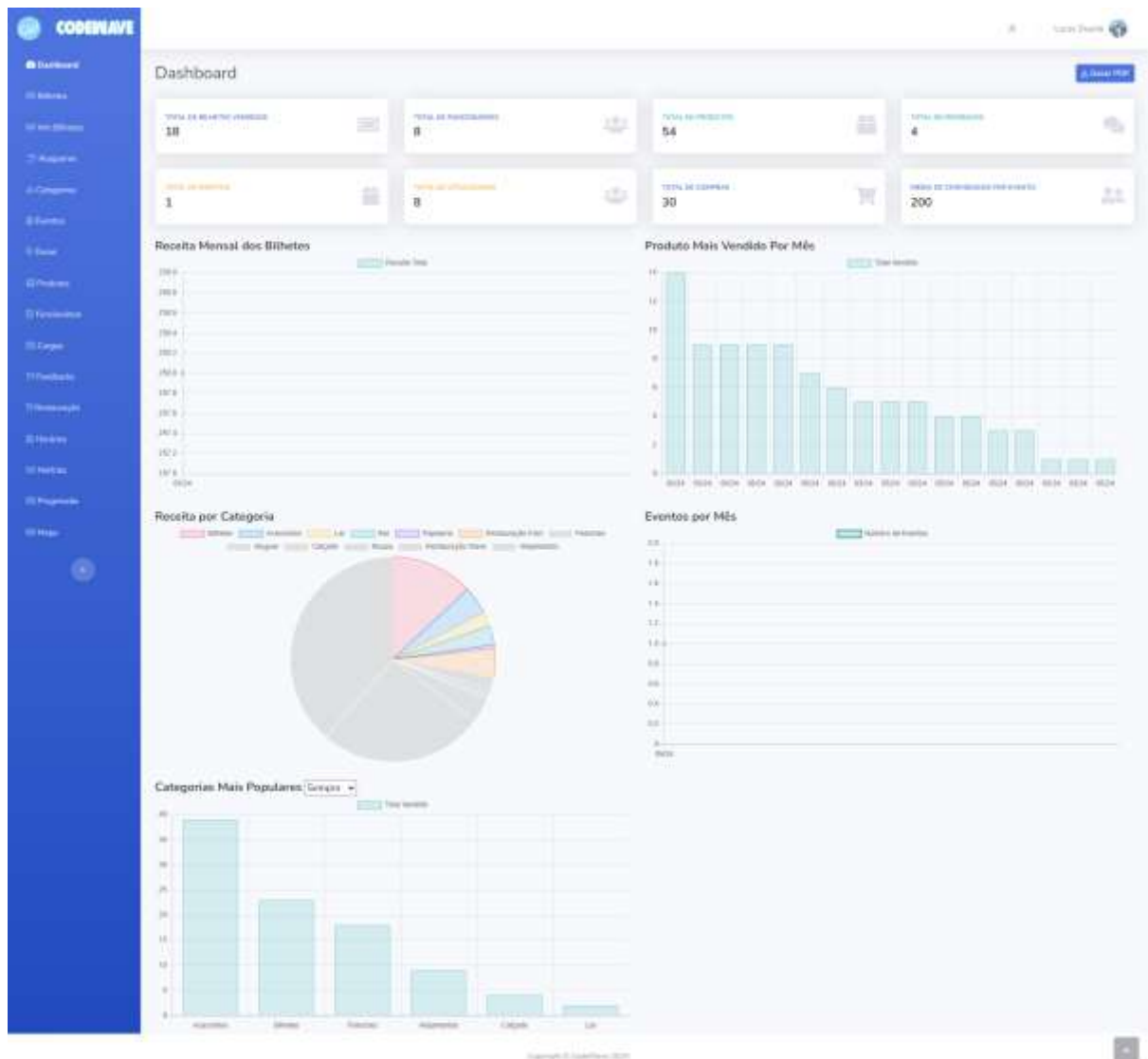


FIGURA 55 - DASHBOARD ADMIN

12.17 Dashboard – Bilhetes

A Dashboard Bilhetes permite que os bilhetes sejam validados. Consoante a quantidade de bilhetes que foram ativados nesse dia, a informação da lotação e o número de bilhetes disponíveis vão sendo alteradas. Essa informação é importante para saber a lotação do parque, não causando vendas excessivas e pondo em causa a segurança do parque.

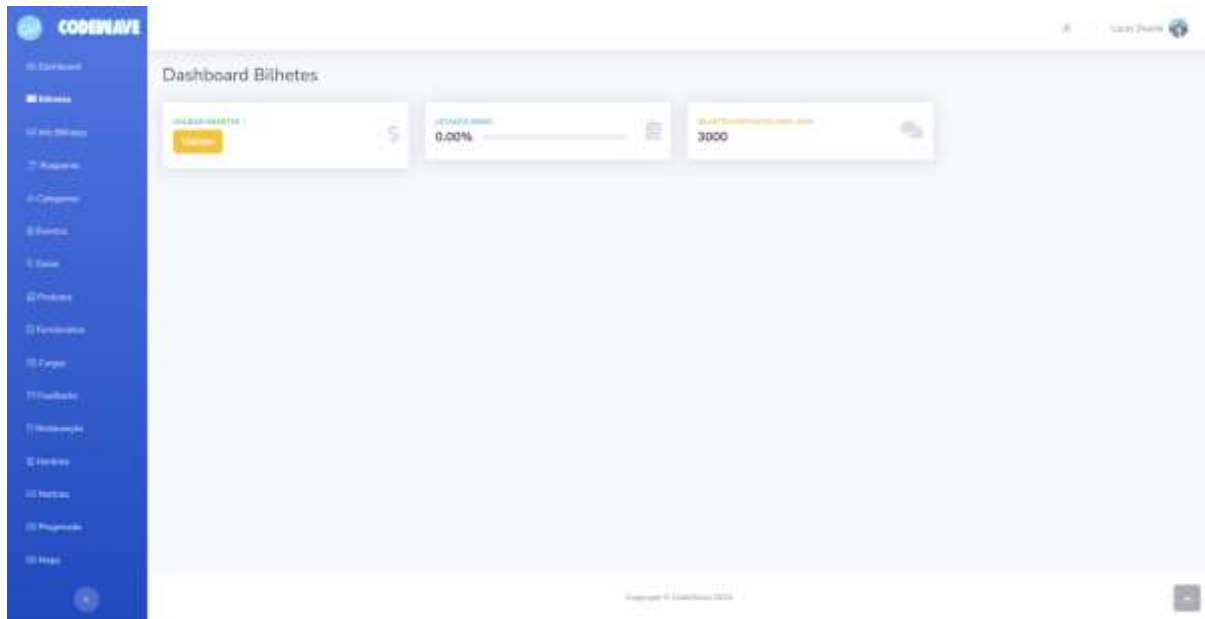
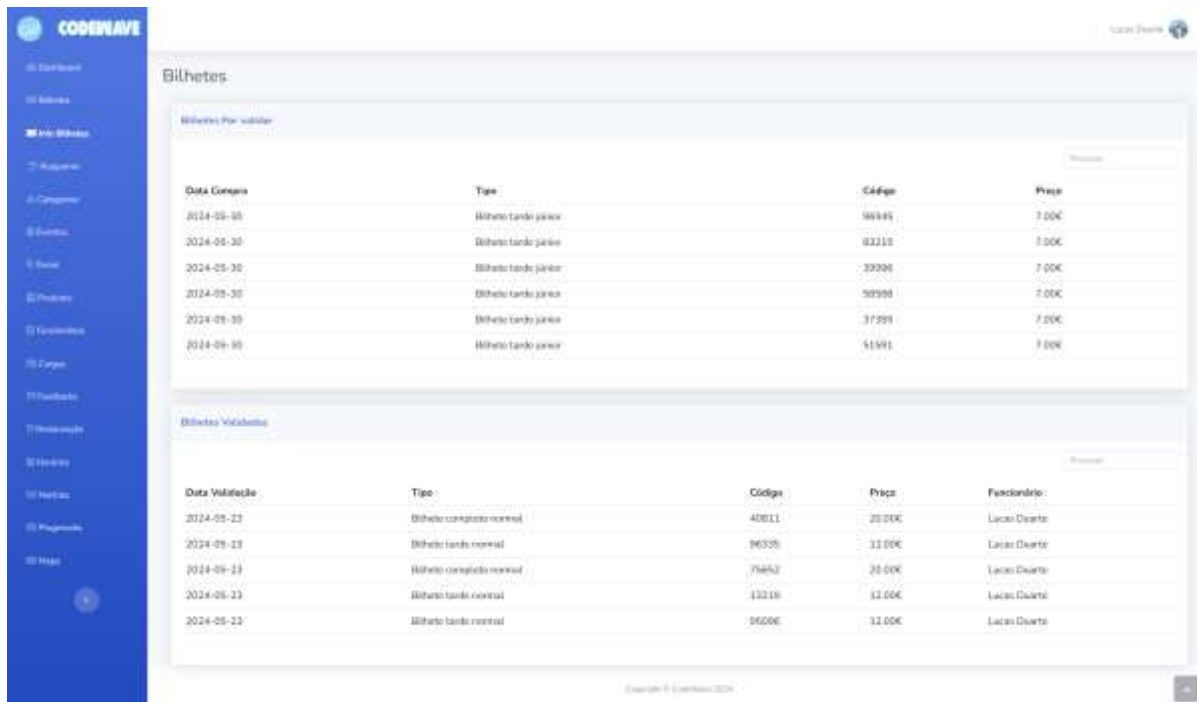


FIGURA 56 - DASHBOARD BILHETES

12.18 Dashboard – Info Bilhetes

A Dashboard Info Bilhete mostra ao Admin todos os bilhetes que estão por validar e os que já foram validados, assim como a data para quando é o bilhete.



Bilhetes:

Bilhetes Por Validar

Data Concreta	Tipo	Código	Preço
2024-09-30	Bilhete tanto já vai	98845	7.00€
2024-09-30	Bilhete tanto já vai	83210	7.00€
2024-09-30	Bilhete tanto já vai	39906	7.00€
2024-09-30	Bilhete tanto já vai	98988	7.00€
2024-09-30	Bilhete tanto já vai	37299	7.00€
2024-09-30	Bilhete tanto já vai	51591	7.00€

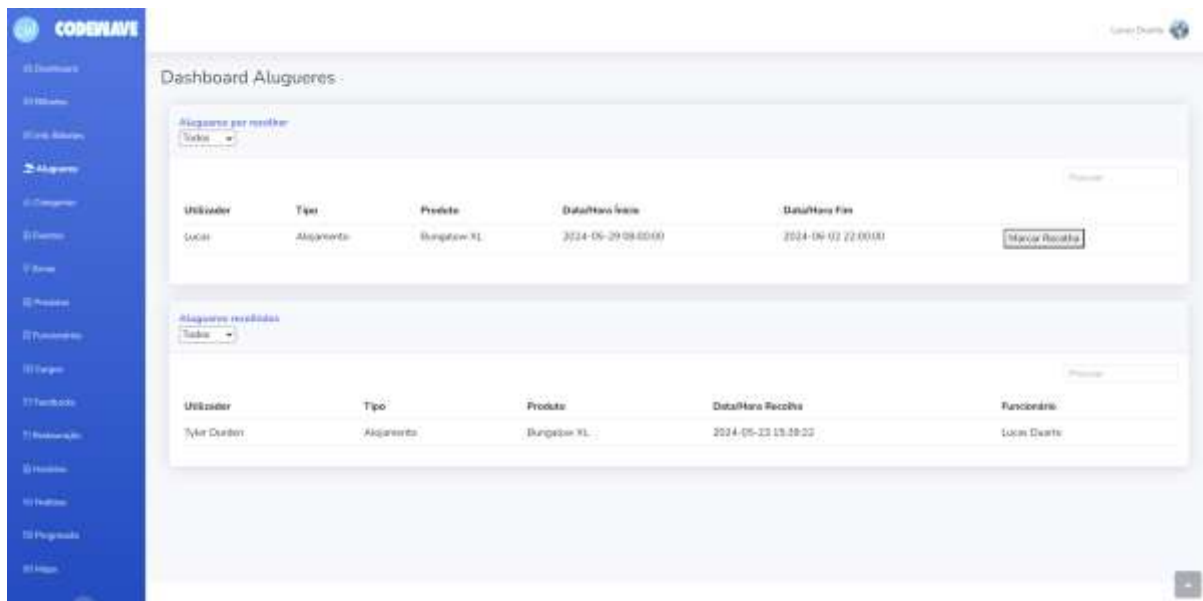
Bilhetes Validados

Data Validação	Tipo	Código	Preço	Funcionário
2024-09-23	Bilhete completo normal	40811	20.00€	Luís Duarte
2024-09-23	Bilhete tanto normal	96335	12.00€	Luís Duarte
2024-09-23	Bilhete completo normal	75652	20.00€	Luís Duarte
2024-09-23	Bilhete tanto normal	13218	12.00€	Luís Duarte
2024-09-23	Bilhete tanto normal	96606	12.00€	Luís Duarte

FIGURA 57 - DASHBOARD INFO BILHETES

12.19 Dashboard – Alugueres

Na Dashboard Alugueres, temos duas tabelas, uma que marca os alugueres por “recolher” e outra com os “recolhidos”. Desta forma, o funcionário ou o gerente pode confirmar o aluguer através de um código que é enviado para o email do utilizador que fez o aluguer. Ao inserir o código que o utilizador recebeu no email, o aluguer é confirmado e passa automaticamente para a tabela de “recolhidos”.



Dashboard Alugueres

Alugueres por recolher

Utilizador	Tipo	Produto	Data/Hora Inicio	Data/Hora Fim
Luca	Alugueres	Burgster XL	2024-05-29 08:00:00	2024-06-02 22:00:00

Alugueres recolhidos

Utilizador	Tipo	Produto	Data/Hora Recolha	Funcionário
Tyler Darden	Alugueres	Burgster XL	2024-05-23 13:38:22	Luca Darden

FIGURA 58 - DASHBOARD ALUGUERES

12.20 Dashboard – Categorias

Na Dashboard Categorias, para além das operações de adicionar, editar e apagar categorias, o gerente pode marcar cada categoria como “ativa”, desta forma todos os produtos desta categoria irão ser mostrados na página da loja online.

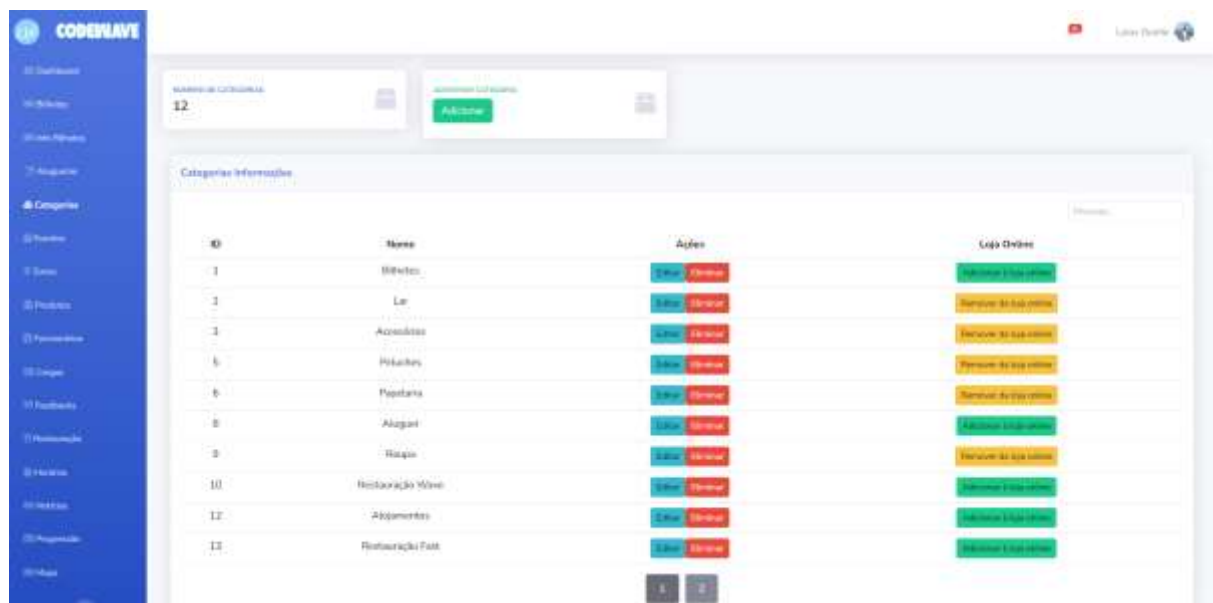
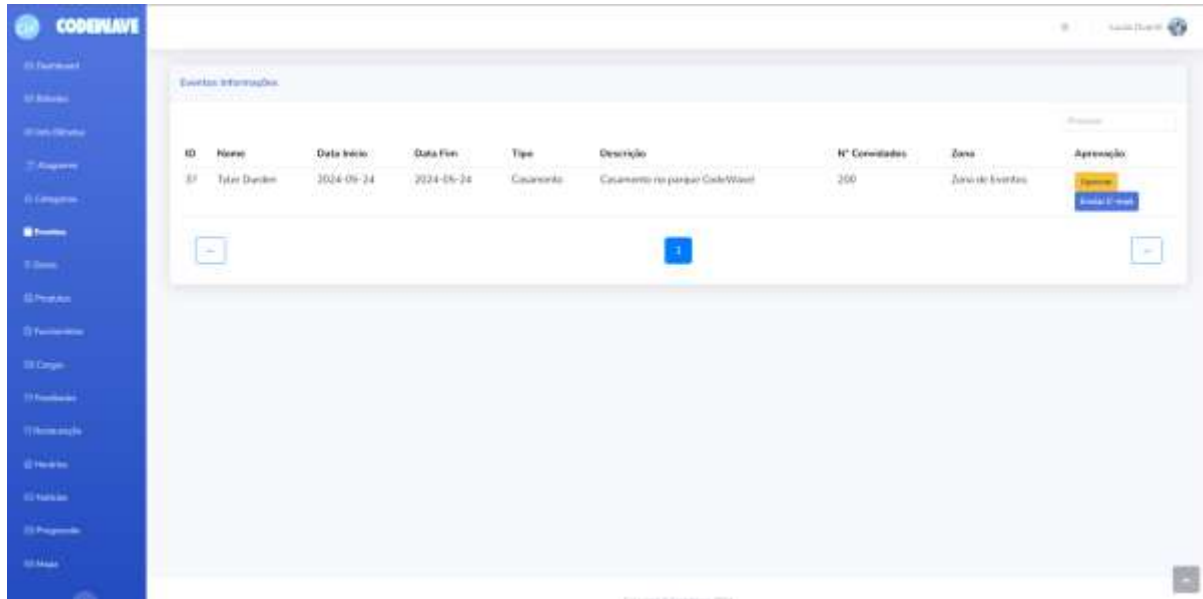


FIGURA 59 - DASHBOARD CATEGORIAS

12.21 Dashboard – Eventos

Na Dashboard Eventos, o Gerente pode ver todos os pedidos de eventos que os utilizadores fizeram. Antes de aprovar o evento, o Gerente pode entrar em contacto com o utilizador para esclarecer mais aprofundadamente os detalhes do evento.

Assim que todos os detalhes forem esclarecidos e houver disponibilidade pode clicar no botão “Aprovar” que irá passar de amarelo para verde (Aprovado).



ID	Nome	Data Inicio	Data Fim	Tipo	Descrição	N.º Convidados	Zona	Ação
33	Tyler Darden	2024-06-24	2024-06-24	Casamento	Casamento no parque CodeWave	200	Zona de Eventos	Apagar Atualizar Evento

FIGURA 60 - DASHBOARD EVENTOS

12.22 Dashboard – Zonas

Na Dashboard Zonas, o Gerente pode adicionar novas zonas ao parque, editar, ou apagar as mesmas.

As zonas do parque, servem para definir o sítio dos alugueres, dos eventos ou de outra atividade relativa ao parque.

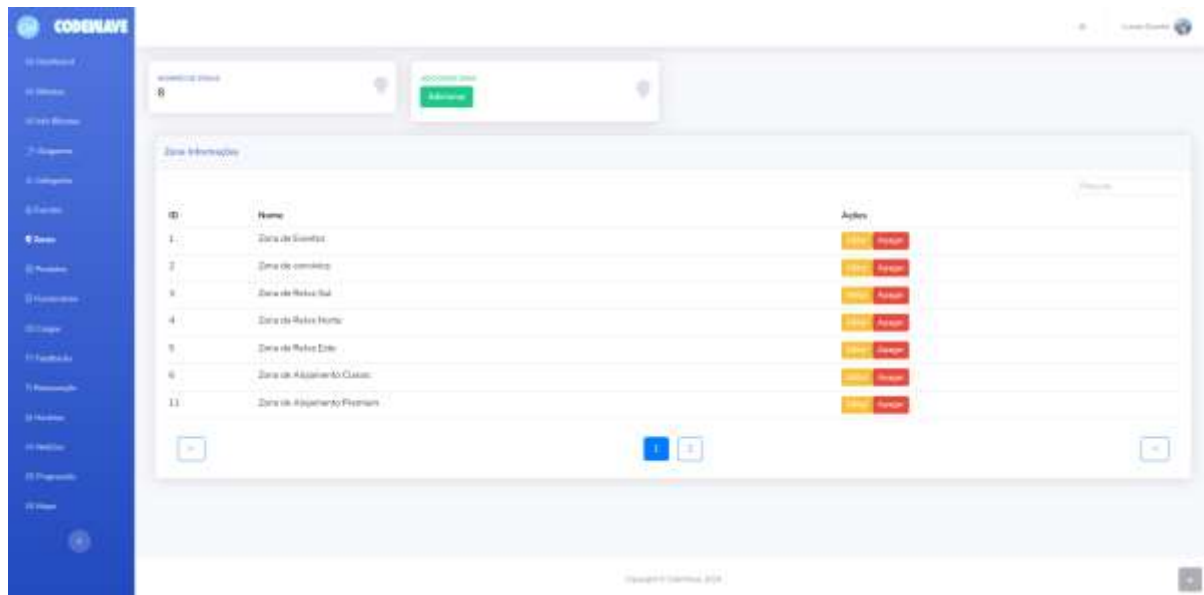


FIGURA 61 - DASHBOARD ZONAS

12.23 Dashboard – Produtos

Na dashboard Produtos é possível visualizar todos os produtos disponíveis, além disso, é possível editar as suas informações ou mesmo apagar o produto.

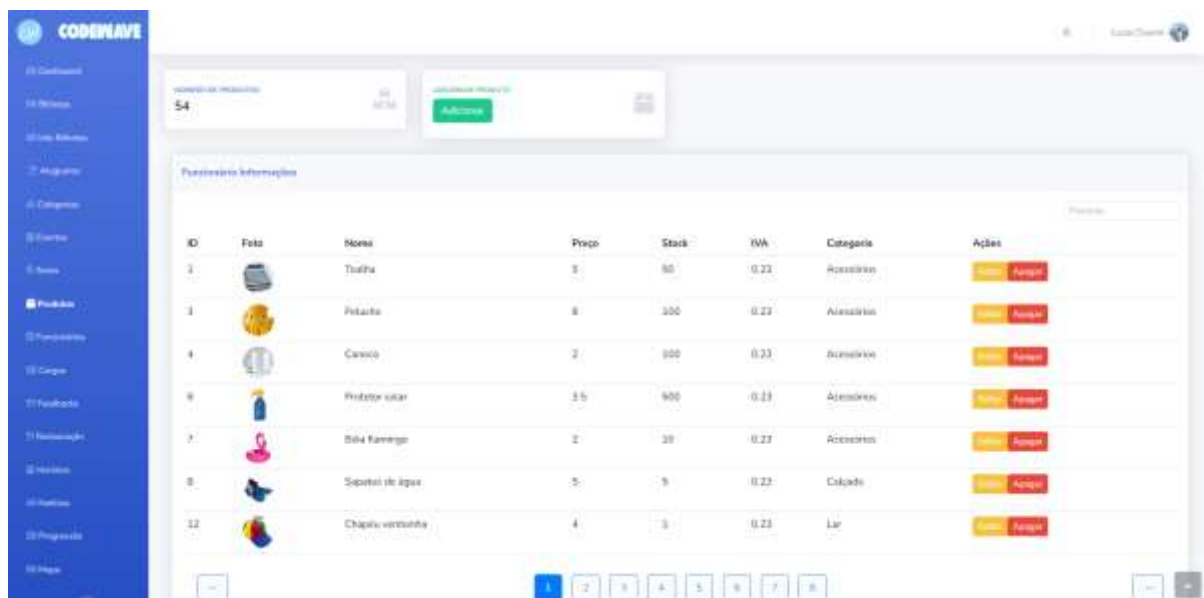


FIGURA 62 - DASHBOARD PRODUTOS

12.24 Dashboard – Funcionários

Na Dashboard Funcionários, o gerente pode adicionar um funcionário, incluindo a sua foto e cargo. O email e as passwords definidas pelo gerente no funcionário serão usadas para autenticar os funcionários nas dashboards.

Também é possível editar as informações de cada funcionário, caso seja necessário. A função de apagar funcionários está disponível caso haja um engano ou o funcionário já não pertença há entidade patronal.

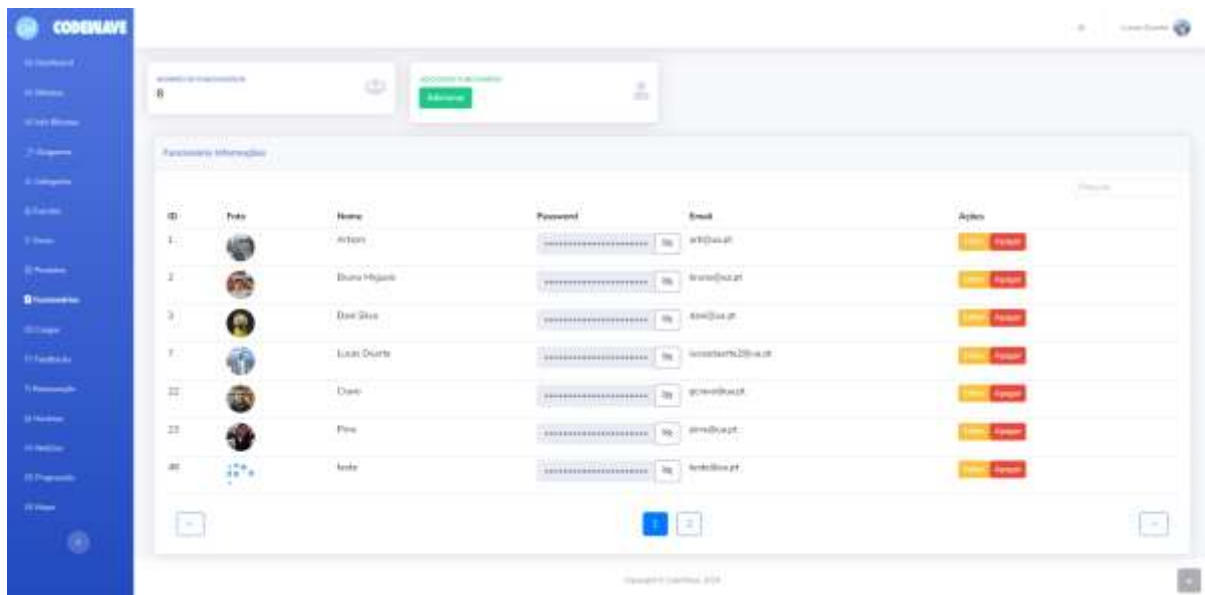


FIGURA 63 - DASHBOARD FUNCIONÁRIOS

12.25 Dashboard – Cargos

Na Dashboard Cargos, o gerente pode realizar 3 operações, sendo elas, editar um cargo, adicionar ou apagar. Ao adicionar um cargo aparece um pop-up para preencher o nome do cargo.

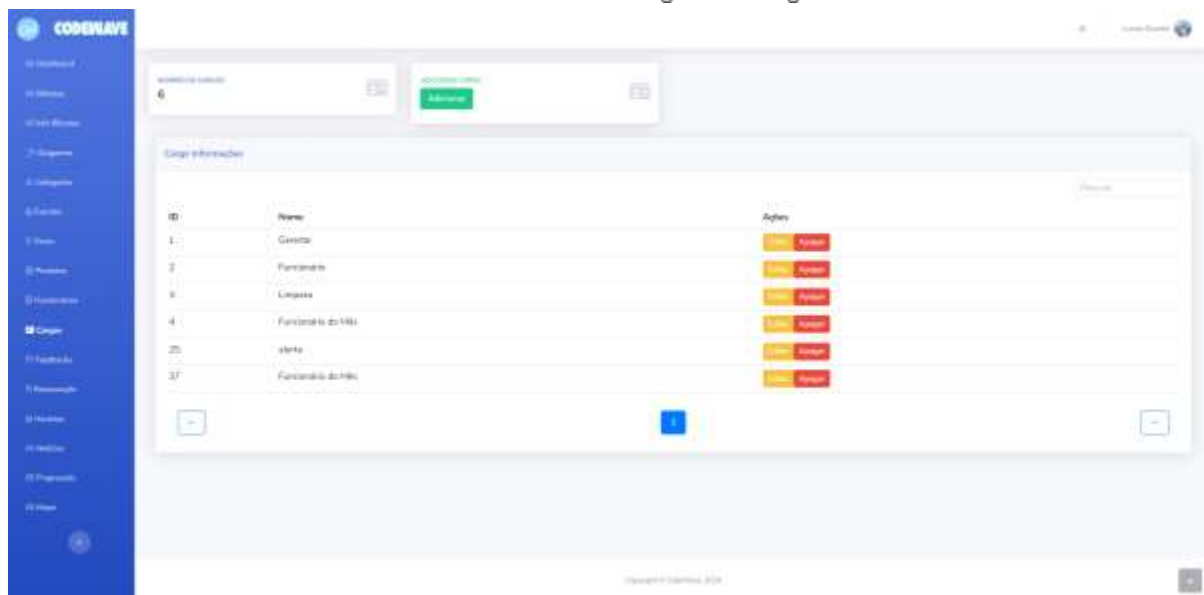


FIGURA 64 - DASHBOARD CARGOS

12.26 Dashboard – Feedbacks

Na Dashboard Feedbacks o gerente, pode ver os feedbacks que os utilizadores deram no website, assim como a sua classificação, e apagar os comentários com conteúdo inadequado ou irregular.

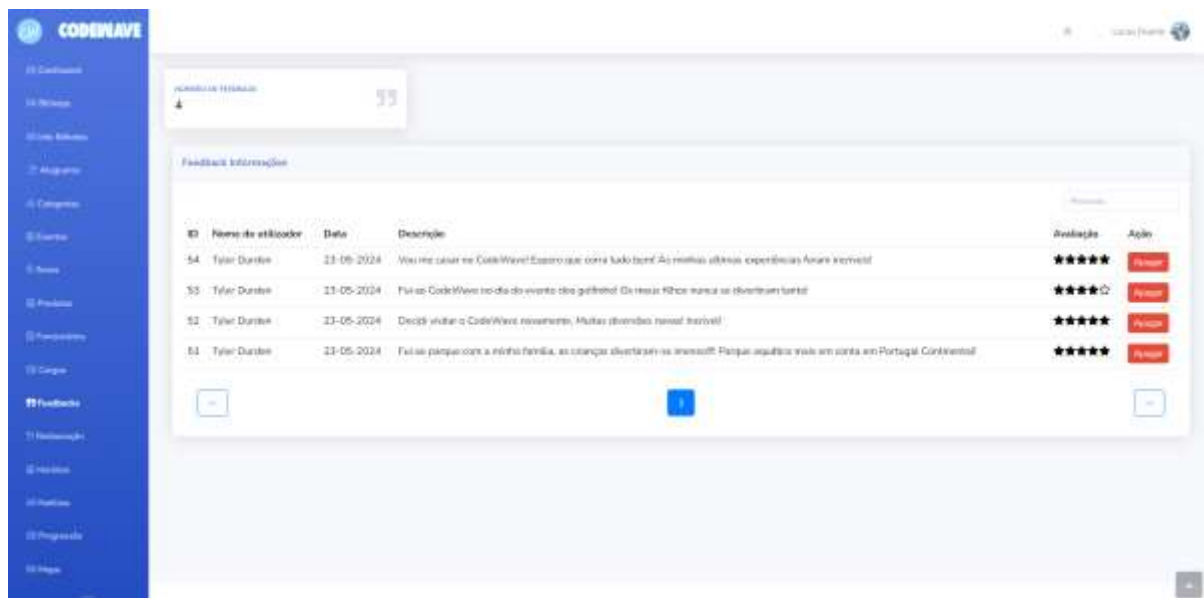


FIGURA 65 - DASHBOARD FEEDBACKS

12.27 Dashboard – Restauração

Na Dashboard Restauração, podemos realizar 3 operações, como editar um produto, adicionar um produto ou apagar o mesmo.

Ao adicionar um produto para além das informações base do produto, devemos escolher se vai estar associado ao Restaurante Wave, Fast ou ao Bar do parque. Para complementar, e facilitar a visualização do que se trata o produto, podemos dar upload de uma imagem para associar ao mesmo.

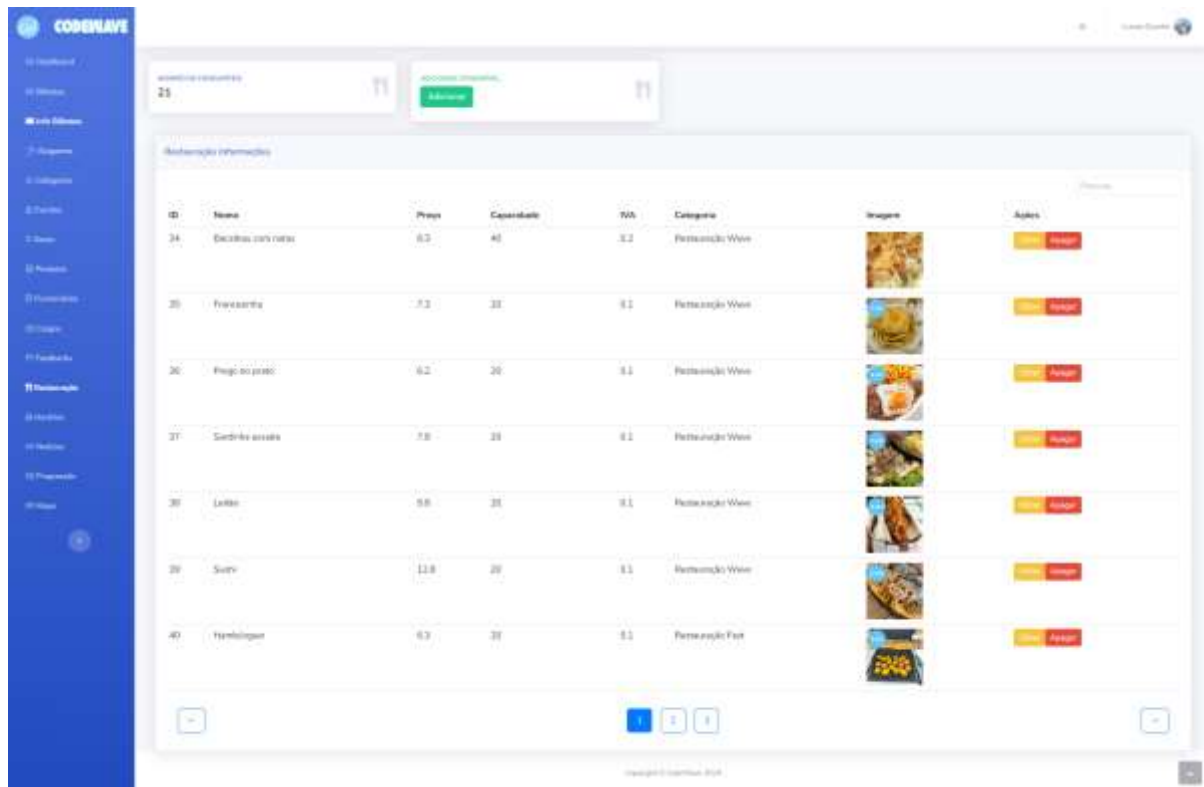


FIGURA 66 - DASHBOARD RESTAURAÇÃO

12.28 Dashboard – Horários

A Dashboard Horários permite a visualização e realizar a alteração dos horários do parque.



FIGURA 67 - DASHBOARD HORÁRIOS

12.29 Dashboard – Notícias

Na Dashboard Notícias, é possível que o admin edite ou apague notícias existentes e que estejam disponíveis no website.

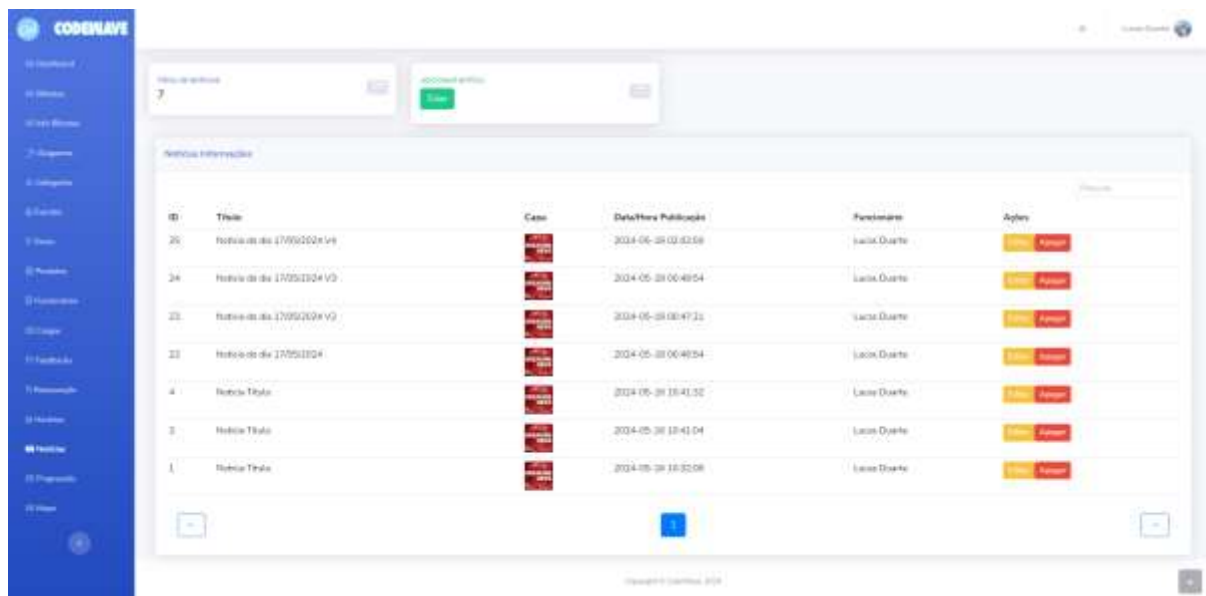
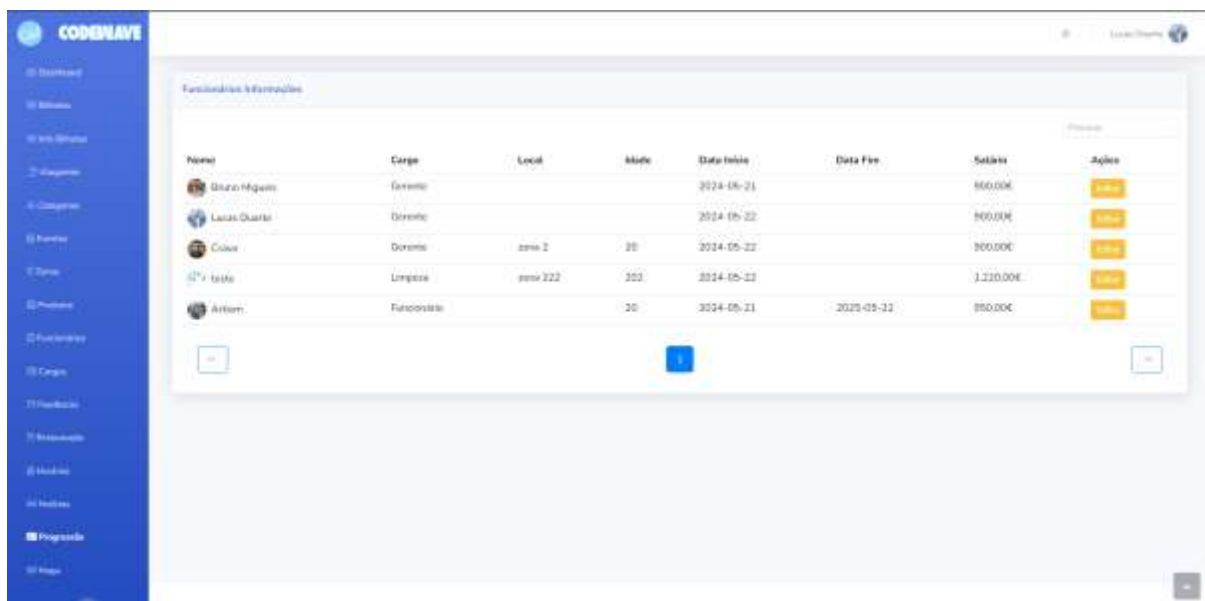


FIGURA 68 - DASHBOARD NOTÍCIAS

12.30 Dashboard – Progressão

Na Dashboard Progressão, o gerente pode atribuir a cada funcionário informações extras, como a idade, salário e o local em que se encontra colocado (zona de trabalho). As datas de Início e Fim são para ter um registo de quando o funcionário foi contratado e até quando ficou na empresa.

Para mudar as informações de cada funcionário, devemos clicar no botão “Editar” do funcionário que pretendemos, e editar ou acrescentar as respetivas informações.



The screenshot shows a web application interface with a sidebar menu on the left and a main content area. The sidebar menu includes options like Dashboard, Relatórios, Funcionários, and Progressão. The main content area displays a table titled 'Funcionários e Informações' with columns for Nome, Cargo, Local, Idade, Data Início, Data Fim, Salário, and Ações. The table contains five rows of employee data, each with an 'Editar' button in the Ações column.

Nome	Cargo	Local	Idade	Data Início	Data Fim	Salário	Ações
Guilherme Miguel	Gerente			2024-08-21		800.00€	Editar
Luís Duarte	Gerente			2024-08-22		800.00€	Editar
Cláudia	Gerente	zona 2	30	2024-05-22		900.00€	Editar
João Silva	Emprego	zona 222	302	2024-05-22		1.220.00€	Editar
Artur	Funcionário		30	2024-05-21	2025-05-22	850.00€	Editar

FIGURA 69- DASHBOARD PROGRESSÃO

12.31 Dashboard – Mapa

Caso o gerente deseje ligar ou controlar a iluminação do parque aquático, é possível que isso seja feito através do website. Através do login como gerente é possível ligar a totalidade ou selecionar individualmente as luzes a ligar. Isto permite ao gerente ter um controlo total e em qualquer dispositivo.

Num mercado cada vez mais orientado para a tecnologia, a implementação de sistemas inovadores pode ser crucial para melhorar a eficiência operacional e oferecer uma experiência mais fluida tanto para os funcionários. quanto para os clientes

Como funciona:

O website tem conexão com um controlador que está ligado ao sistema de luzes do parque aquático, esse controlador quando recebe ordem de ligar/desligar luz do website vai realizar essa ordem sem ser preciso estar alguém presente no parque.

Vai permitir iluminar rapidamente sem deslocação a secção necessária do parque como por exemplo em caso de emergência.

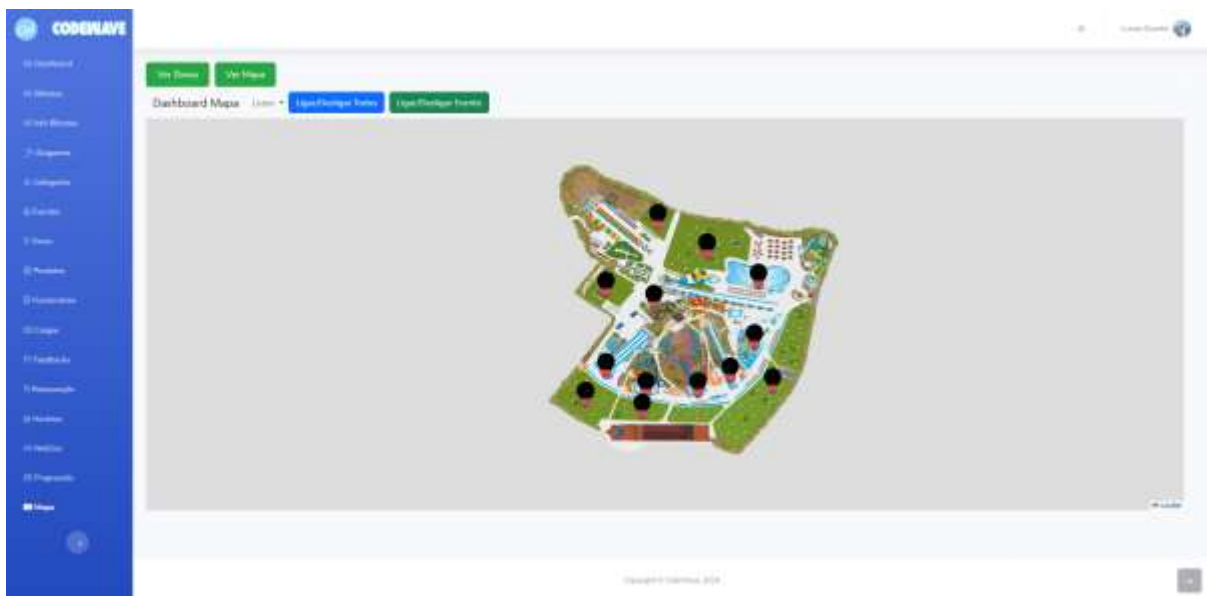


FIGURA 70 - DASHBOARD ZONAS

13. Mapa de AR (Realidade Aumentada)

Desenvolvemos um cenário com o mapa do parque em realidade aumentada, utilizando o Adobe Aero, que inicialmente estava projetado para ser exibido em 3D; contudo, devido a limitações técnicas, conseguimos implementar apenas a versão 2D.

Este mapa permite aos utilizadores explorar diversas funcionalidades, incluindo links rápidos para facilitar a navegação pelo site. Além disso, oferece informações adicionais, como links para determinadas funções e algumas animações simples que melhoram a experiência de uso do utilizador.

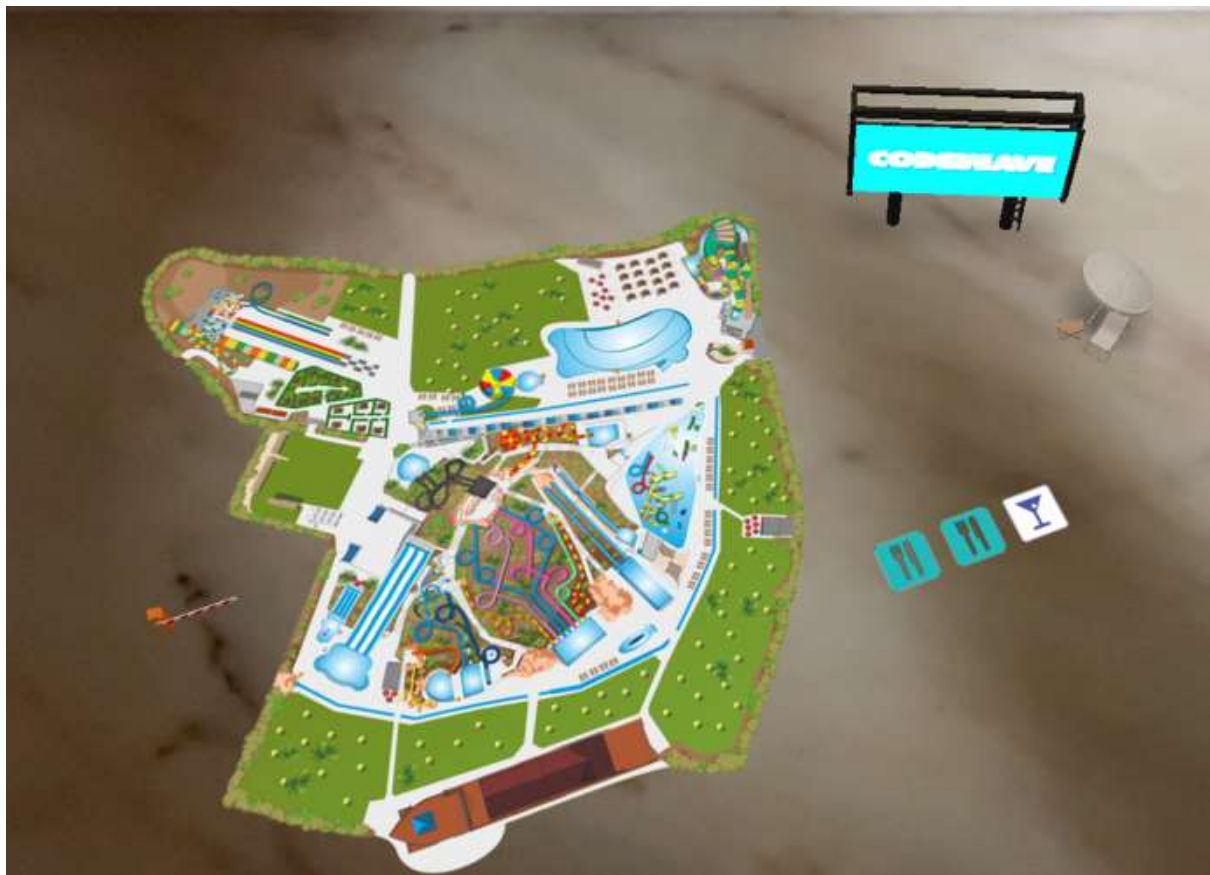


FIGURA 71 - MAPA REALIDADE AUMENTADA

14. Controlador do Parque de Estacionamento

Este diagrama [Figura 72 - Diagrama de estacionamento do parque (Arduino)] ilustra um circuito para controlar o estacionamento do parque, utilizando um Arduino Uno, dois sensores PIR (Passive Infrared) e um servo motor. Os sensores PIR são usados para detetar a entrada e saída de carros no parque, enquanto o servo motor seria utilizado para controlar uma cancela da entrada/saída do parque.

As leituras dos sensores PIR são enviadas para uma base de dados, onde uma dashboard exibe o número de lugares disponíveis em tempo real. No entanto, não foi possível implementar este sistema completamente devido ao tempo excessivo necessário para que os dados chegassem à base de dados e retornassem, causando atrasos significativos na atualização da disponibilidade de lugares.

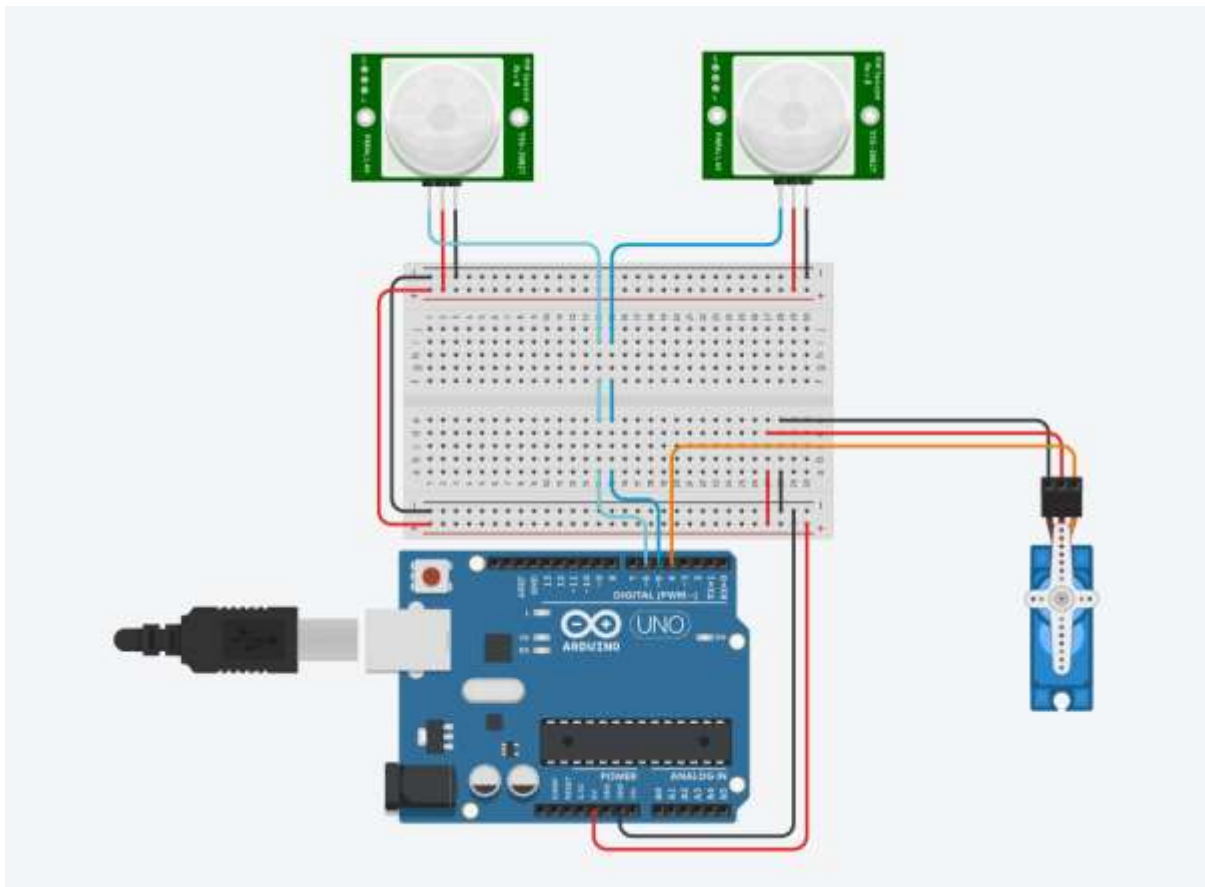


FIGURA 72 - DIAGRAMA DE ESTACIONAMENTO DO PARQUE (ARDUINO)

15. Controlador de iluminação Arduino

Este diagrama [Figura 73 - Diagrama de iluminação (Arduino)] mostra um circuito usando uma placa Arduino Uno para controlar LEDs, que representam as luzes do parque. Os pinos digitais do Arduino estão conectados aos ânodos dos LEDs através de resistências, que limitam a corrente para proteger os LEDs e o Arduino.

O código no Arduino está programado para acender os LEDs em sequência, consoante o ‘status’ (true; false) que está na Base de Dados na tabela Luzes. Este sistema é controlado através de uma Dashboard do site, permitindo que o gerente, ajuste as luzes do parque remotamente, demonstrando um controlo básico de saídas digitais integrado com uma interface web.

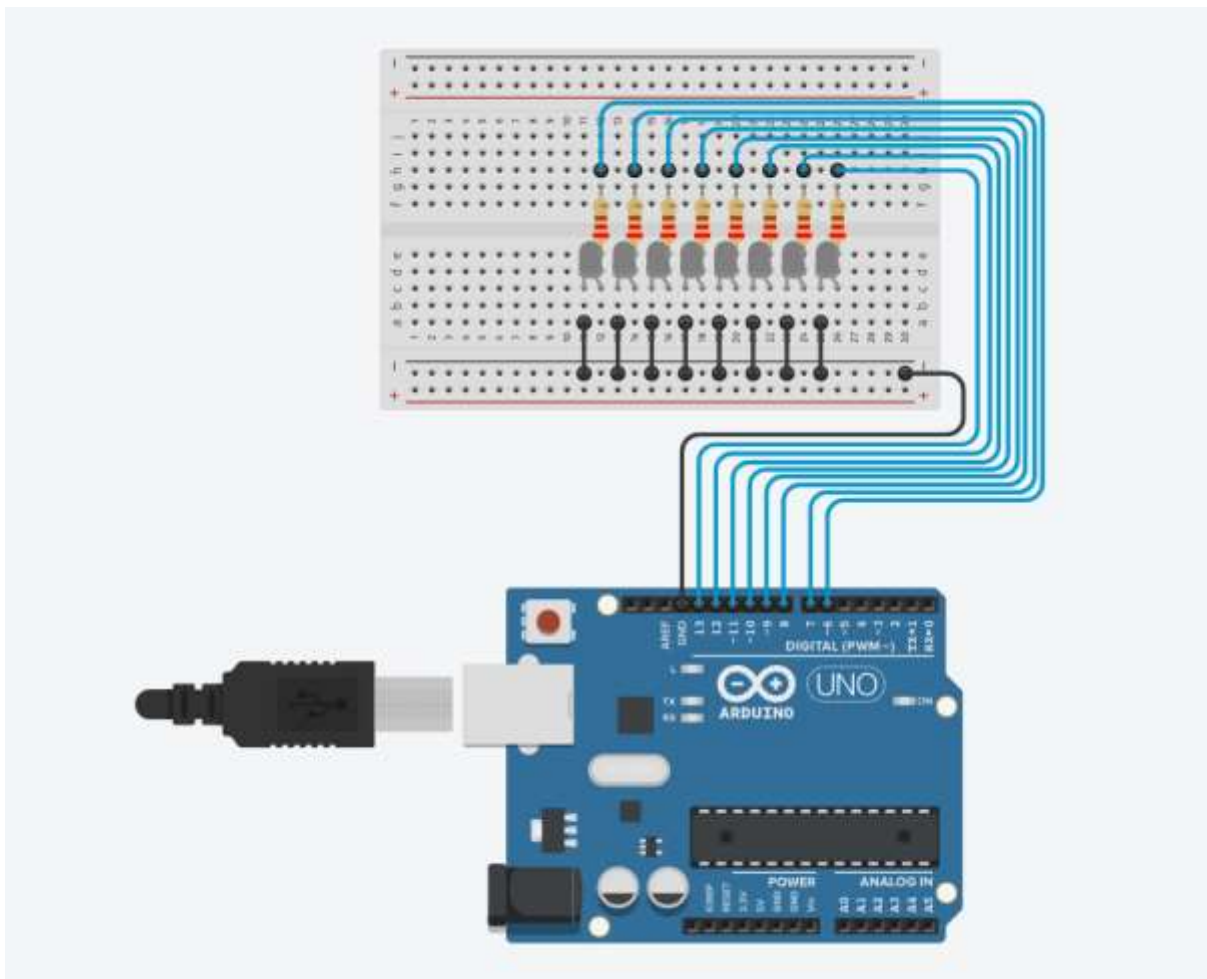


FIGURA 73 - DIAGRAMA DE ILUMINAÇÃO (ARDUINO)

```
const int num_leds = 13;
int leds[num_leds] = {53, 51, 49, 47, 45, 43, 41, 39, 37, 35, 33, 31, 52};
bool led_states[num_leds] = {false};

void setup() {
    Serial.begin(9600);

    for (int i = 0; i < num_leds; i++) {
        pinMode(leds[i], OUTPUT);
    }
}

void loop() {
    if (Serial.available() > 0) {
        int led_num = Serial.parseInt();
        if (led_num >= 1 && led_num <= num_leds) {
            String command = Serial.readStringUntil('\n');
            Serial.print("Comando recebido para LED ");
            Serial.print(led_num);
            Serial.print(": ");
            Serial.println(command);

            if (command == "True") {
                led_states[led_num - 1] = true;
                Serial.println("Ligando LED");
            } else if (command == "False") {
                led_states[led_num - 1] = false;
                Serial.println("Desligando LED");
            }
        }
    }

    for (int i = 0; i < num_leds; i++) {
        digitalWrite(leds[i], led_states[i]);
    }
}
```

FIGURA 74 - CÓDIGO CONTROLADOR DE LEDS

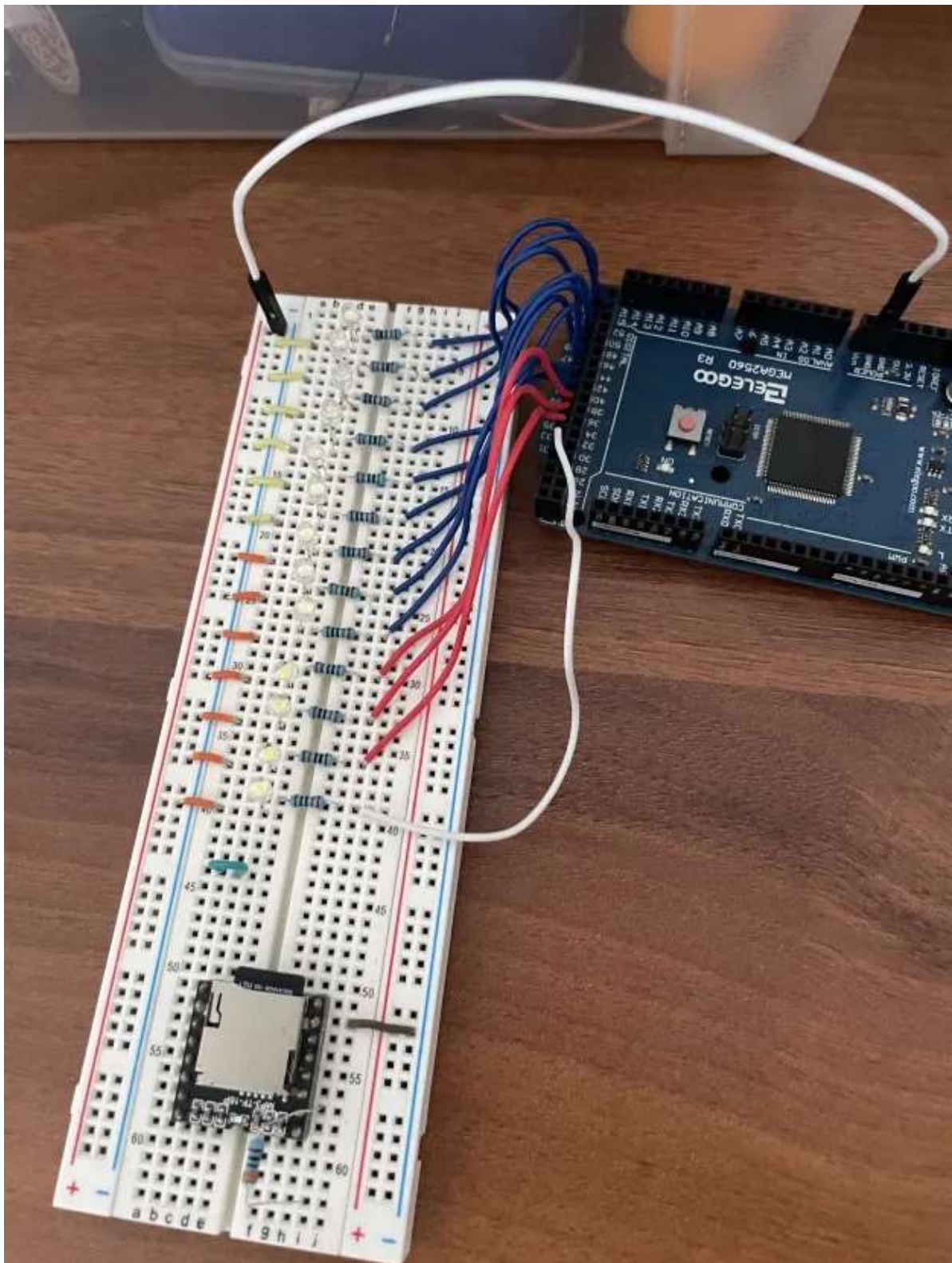


FIGURA 75 - PROTÓTIPO DO CONTROLADOR DE ILUMINAÇÃO

16. Testes e validação

Durante o processo de desenvolvimento de *software*, os testes desempenharam um papel muito importante. Estes são essenciais para avaliar a qualidade da aplicação web e garantir que ele atenda os requisitos do utilizador.

A importância dos testes verifica-se na identificação antecipada de defeitos e falhas no programa. Contribuindo assim para a confidencialidade e segurança do software, bem como para o cumprimento das exigências do utilizador. Além disso, no contexto de projetos profissionais, os testes podem reduzir os custos de desenvolvimento, uma vez que permitem a correção antecipada de defeitos.

Diferentes tipos de testes de *software* foram aplicados. No projeto em questão, realizamos testes de sistema para verificar o funcionamento adequado dos diversos componentes de *software* quando combinados. Para além disso, realizamos testes de validação para garantir que o *software* atendessem aos requisitos dos utilizadores. Foram realizados também testes de usabilidade que foram importantes para avaliar a experiência do utilizador, assegurando que o website fosse intuitivo e que proporcionasse uma interação satisfatória.

16.1 Testes de sistema e de validação

Durante o desenvolvimento de *software*, os testes de sistema desempenham um papel crucial. Estes testes são realizados no sistema como um todo, ao verificar se ele satisfaz todos os requisitos. Os *testers* avaliam tanto os requisitos funcionais quanto os não funcionais após a integração dos módulos e componentes individuais. O teste de sistema é uma categoria de teste de caixa preta, pois focam-se exclusivamente nas características funcionais externas do *software*. Os *testers* não requerem conhecimento detalhado da programação ou da estrutura do código da plataforma web para avaliar o *software* durante os testes de sistema. Ao invés disso, eles avaliam o desempenho do *software* sob a perspetiva de um utilizador.

Além disso, foram realizados testes de validação, pois estes são essenciais para garantir que o *software* atenda aos requisitos dos utilizadores e do negócio. Eles avaliam se o *software* neste caso o site para examinar se é adequado para o uso pretendido e se atende às necessidades dos utilizadores. Os testes de validação também verificam se o *software* está pronto para ser apresentado ao cliente. Essa etapa é fundamental no processo de desenvolvimento, pois ajuda a garantir que o *software* corresponde às expectativas iniciais.

No contexto específico do projeto, também realizamos testes de desempenho para garantir que a base de dados conseguisse manipular os dados sem problemas quando o sistema estivesse sob maior carga. Estes testes garantem que o tempo de resposta do sistema permanece dentro dos parâmetros aceitáveis.

No geral estes testes foram realizados pelos membros do grupo, principalmente na fase final do projeto, para assegurar que a plataforma *web* respondesse às necessidades dos utilizadores e do negócio.

16.2 Testes de usabilidade

A avaliação de usabilidade de uma aplicação pode ser realizada por meio de vários métodos. No nosso caso, optamos pelo teste de usabilidade. Este método envolve a avaliação de um produto ou serviço com a participação de utilizadores finais representativos. Durante estes testes, os participantes são convidados a realizar uma série de tarefas típicas, enquanto os responsáveis pela avaliação observam, ouvem e anotam. O objetivo principal desses testes é identificar problemas relacionados com a usabilidade, recolher dados qualitativos e quantitativos e determinar o nível de satisfação dos participantes com o produto ou serviço. Uma das vantagens desses testes é a deteção antecipada de falhas de usabilidade. Portanto, esse método foi escolhido para garantir a qualidade do nosso produto.

Para conduzir um teste de usabilidade com sucesso, é necessário contar com alguns documentos essenciais:

- Grelha de tarefas a realizar e observações;
- Documento de consentimento de participação no teste [**Figura 77** - Documento de consentimento de participação no teste];
- Questionário de satisfação global [**Figura 78** - Questionário de satisfação];

Tarefa		Execução			Erros	Dificuldades	Dúvidas	Observações	Duração
#	Descrição	Bastante Dificuldade	Alguma Dificuldade	Sem Dificuldade					
Tarefas como funcionário									
1	Verificar o horário de funcionamento do parque								
2	Perguntar ao assistente de IA sobre o horário do parque								
3	Realizar a compra de algo na loja								
4	Ver histórico de compras								
5	Visitar o mapa do parque								
6	Alugar um Bungalow por 4 dias								

TABELA 53 TESTES DE USABILIDADE FUNCIONÁRIO

Tarefa		Execução			Erros	Dificuldades	Dúvidas	Observações	Duração
#	Descrição	Bastante Dificuldade	Alguma Dificuldade	Sem Dificuldade					
Tarefas como Admin									
1	Fazer login como <i>Admin</i>								
2	Editar uma zona do parque								
3	Desligar a iluminação do parque								
4	Editar algo na área de restauração								
5	Fazer <i>download</i> do PDF das estatísticas								
6	Eliminar categoria								

TABELA 54 - TESTES DE USABILIDADE GERENTE

Participação em teste de usabilidade

No âmbito do Projeto Temático em Aplicações Web do Módulo Temático em Aplicações Web lecionada na Escola Superior de Tecnologia e Gestão de Águeda, está-se a desenvolver um website para um parque aquático.

O teste de usabilidade tem como objetivo testar as funcionalidades do protótipo desenvolvido, reforçando desde já que este é um teste à interface e não ao utilizador. Para tal, irá realizar tarefas no mesmo e responder a um breve questionário.

O seu contributo é extremamente relevante para este projeto e garante-se o anonimato de todas as informações de carácter pessoal.

Obrigado pela sua participação.

Consentimento informado

Deste modo, declaro ter compreendido a explicação que me foi dada sobre o projeto a decorrer e as informações recolhidas são anónimas. Eu entendo que os resultados do projeto podem ser publicados em revistas científicas, apresentados em reuniões/eventos científicos e utilizados em atividades de investigação, sem qualquer violação de confidencialidade/anonimato. Ao participar nesta atividade, autorizo o uso de dados anónimos ara a finalidade do projeto que lhe está associada e mencionada acima.

Águeda, de de

Assinatura: _____

FIGURA 77 - DOCUMENTO DE CONSENTIMENTO DE PARTICIPAÇÃO NO TESTE

1. Gostaria de utilizar este sistema com frequência *

1	2	3	4	5
---	---	---	---	---

Discordo
completamenteConcordo
completamente

6. Acho que o sistema apresenta muita inconsistência *

1	2	3	4	5
---	---	---	---	---

Discordo
completamenteConcordo
completamente

2. O sistema é desnecessariamente complexo *

1	2	3	4	5
---	---	---	---	---

Discordo
completamenteConcordo
completamente

7. Penso que a maioria das pessoas aprenderia rapidamente como utilizar o sistema *

1	2	3	4	5
---	---	---	---	---

Discordo
completamenteConcordo
completamente

3. Achei o sistema fácil de utilizar *

1	2	3	4	5
---	---	---	---	---

Discordo
completamenteConcordo
completamente

8. Achei que o sistema era confuso de utilizar *

1	2	3	4	5
---	---	---	---	---

Discordo
completamenteConcordo
completamente

4. Acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para utilizar o sistema *

1	2	3	4	5
---	---	---	---	---

Discordo
completamenteConcordo
completamente

9. Senti-me confiante ao utilizar o sistema *

1	2	3	4	5
---	---	---	---	---

Discordo
completamenteConcordo
completamente

5. Acho que as várias funções do sistema estão bem integradas *

1	2	3	4	5
---	---	---	---	---

Discordo
completamenteConcordo
completamente

10. Precisei de aprender várias coisas novas antes de conseguir utilizar o sistema. *

1	2	3	4	5
---	---	---	---	---

Discordo
completamenteConcordo
completamente

FIGURA 78 - QUESTIONÁRIO DE SATISFAÇÃO

Os ANEXO A e ANEXO B contêm as grelhas de observação e o levantamento do questionário, respetivamente. Estes anexos são essenciais para entender a usabilidade do nosso sistema. Nas grelhas de observação, é possível verificar os erros, dúvidas e dificuldades que os utilizadores sentiram ou detetaram ao utilizar o nosso sistema. Estes problemas foram identificados e corrigidos, melhorando assim a experiência do utilizador e a eficiência do nosso sistema.

17. Análise de Resultados

Ao analisar a lista de requisitos previamente mencionada verificou-se que foram levantados 45 requisitos no total, sem distinção entre funcionais e não funcionais.

Como se observa no gráfico da [Figura 79 - Gráfico dos requisitos cumpridos, não cumpridos e parcialmente cumpridos] em termos quantitativos pode-se afirmar que foram cumpridos (≈ 80) % dos requisitos iniciais, tendo (≈ 13) % ficado parcialmente cumpridos e (≈ 7) % por implementar.

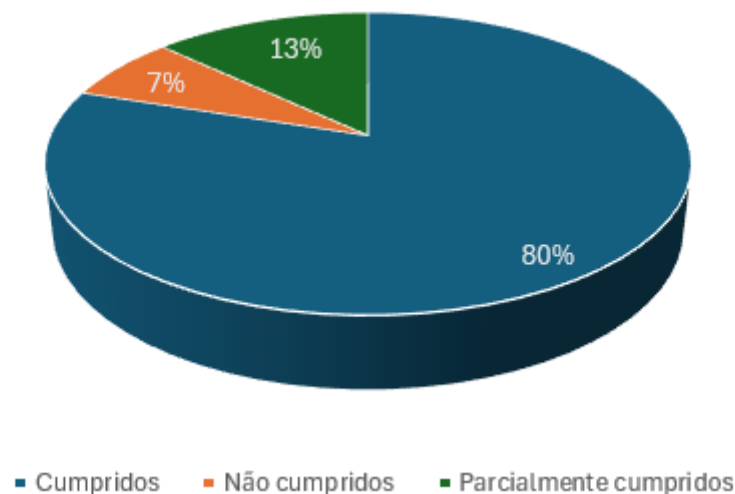


FIGURA 79 - GRÁFICO DOS REQUISITOS CUMPRIDOS, NÃO CUMPRIDOS E PARCIALMENTE CUMPRIDOS

TABELA 55 - REQUISITOS CUMPRIDOS

Requisitos cumpridos

RF.1	Comprar bilhetes
RF.3	Gerir produtos
RF.5	Carrinho de compras
RF.6	Visualização e interatividade do mapa do parque
RF.7	Gestão de reservas para atividades específicas
RF.8	Sistema de notificações e alertas
RF.9	Remover feedbacks
RF.10	Gerir cargos

RF.11 Visualizar horários do parque aquático

RF.12 Integração de API de meteorologia (Open Weather API)

RF.14 Gerir funcionários

RF.16 Agendamento de eventos especiais

RF.17 Gerir restauração

RF.18 Autenticação

RF.19 Validar bilhete

RF.20 Gerir conta

RF.22 Alugar conveniências

RF.23 Gerar relatório de estatísticas do gerente

RF.24 Visualização do mapa do parque em realidade aumentada

RF.25 Comprar produto

RF.26 Enviar feedback

RF.27 Gerir progressão de carreiras

RF.28 Visualizar menus da restauração

RF.29 Gerir horários do parque aquático

RF.30 Visualizar galeria de fotos

RF.31 Contactar o parque aquático

RF.32 Alugar alojamento

RF.33 Ver notícias do parque

RF.34 Gerir notícias do parque

RF.35 Subscrever newsletter

RF.36 Gerir zonas

RF.37 Gerir categorias

RNF.1 Desempenho

RNF.2 Escalabilidade

RNF.3 Usabilidade

RNF.4 Segurança

TABELA 56 - REQUISITOS NÃO CUMPRIDOS

Requisitos não cumpridos

RF.15 Visualizar informações sobre segurança e regras do parque

RNF.5 Backup e Recuperação

RNF.7 Manutenção e Suporte

TABELA 57 - REQUISITOS PARCIALMENTE CUMPRIDOS

Requisitos parcialmente cumpridos

RF.2 Gerir estacionamento

RF.4 Gestão de equipamentos

RF.13 Afluência do parque hora a hora

RF.21 Chat com Integração de IA

RNF.6 Integração

RNF.8 Tempo de Resposta

18. Reflexão Crítica e Conclusões

Nesta secção faz-se uma breve análise crítica sobre todo o trabalho desenvolvido, subdividido em vários pontos. Em primeiro lugar reflete-se sobre as atividades desenvolvidas e estratégias de trabalho adotadas, em seguida sobre o planeamento executado e por fim termina-se com uma síntese final e sugestões de melhoria.

18.1 Atividades Desenvolvidas

O levantamento inicial de requisitos e funcionalidades foi feito com bastantes incertezas inerentes ao funcionamento do sistema. Devido a isto, considerou-se que existiam requisitos insuficientes e não querendo arriscar em algo demasiado simplista foram adicionados requisitos extra de modo a aumentar o conteúdo e a complexidade da aplicação web.

Conclui-se que este projeto e a aplicação resultante do mesmo foram um sucesso. O número de requisitos implementados de importância elevada, assim como as tarefas macro desenvolvidas até então estão na base dos critérios de sucesso utilizados para se poder qualificar o projeto como bem-sucedido.

18.2 Estratégias de Trabalho Adotadas

A estratégia de trabalho adotada, que acabou por ser modelada, resultou das regras de funcionamento de unidades curriculares lecionadas no passado. Também devido a reuniões semanais com discussão de conteúdos e trabalho, que resultariam em planeamento atempado do projeto e dos afazeres, culminando na entrega final deste documento e do website. Este sistema foi bastante vantajoso para a dinâmica do grupo, mesmo havendo alguma disparidade de disponibilidade entre os colegas, pois foram estabelecidas metas semanais para cada um dos elementos, de forma a obter uma distribuição de tarefas igualitária que não prejudicasse nem beneficiasse nenhum membro em específico as tarefas referentes ao desenvolvimento da aplicação foram divididas por funcionalidades, o que necessitou de uma maior dedicação de cada um. A comunicação a entreaajuda e o trabalho árduo foram três pontos muito fortes e importantes, o que se refletiu no produto final.

18.3 Planeamento Previsto e Cronograma Executado

Houve discrepâncias notáveis entre o cronograma programado e a sua real execução, sobretudo no âmbito do tempo necessário à realização de algumas tarefas. Durante as fases de planeamento e desenho do sistema, a contribuição do grupo como um todo foi menos ativa do que o esperado. Esta tendência para um menor envolvimento levou a uma diminuição da capacidade funcional e consequentemente ao atraso de sensivelmente uma semana da entrega intermédia.

Na fase de implementação, o grupo funcionou melhor e estava mais motivado, ao realizar a tarefa "Desenvolvimento do website" de forma detalhada, eficaz e organizada, apesar da distribuição não igualitária nesta fase, o que causou uma sobrecarga adicional a parte do grupo, em relação aos membros restantes, devido a participação restrita de alguns. As tarefas relacionadas à "Implementação da base de dados", "Produção do modelo estrutural" e "Produção do modelo de dados persistente" também acabaram por requerer um tempo de execução superior ao inicialmente previsto, devido a inúmeras correções de erros, falhas e problemas com a conexão à base de dados.

18.4 Sugestões para o Futuro

Como parte do desenvolvimento futuro, fica como objetivo concluir os requisitos que ainda não foram implementados, seja forma total ou parcial. Isso permitirá a construção de uma plataforma web com maior robustez e eficiência. Para além disso, foram detetados alguns aspetos que necessitam de melhorias em versões seguintes:

- ♦ **Tratamento de dados:** A eficiência no tratamento de dados é importante para o desempenho geral da aplicação. É necessário melhorar a validação dos dados antes de os inserir na base de dados. Além disso, a criação de índices para consultas SQL frequentes pode melhorar significativamente o acesso aos dados, por exemplo, na recolha da disponibilidade dos bilhetes.
- ♦ **Sistema de backups:** A possibilidade de perda de dados é um risco que precisa ser reduzido. É aconselhável sempre estabelecer um sistema de *backup* frequente para assegurar a integridade dos dados. Esses backups podem ser utilizados para recuperar os dados em situações de falha do sistema.
- ♦ **Qualidade dos testes:** A realização de testes é essencial para assegurar o correto funcionamento da aplicação, bem como para detetar e corrigir eventuais falhas antecipadamente. Embora tenham sido realizados alguns testes, reconhece-se que tanto a quantidade quanto a qualidade poderiam ser melhoradas, um exemplo disso é realizar os testes

unitários que tinham o objetivo de identificar e corrigir falhas em segmentos específicos de código, garantindo assim a integridade e funcionamento correto das mesmas dentro do sistema como um todo.

- ♦ **Adicionar diversão:** A Interface do gerente, onde pode controlar certas configurações do mapa deve ter um botão para adicionar diversão. Ao clicar, o gerente deve preencher um formulário com a descrição da diversão, fazer upload de uma imagem, escolher um *pin* para o mapa e selecionar no mapa uma localização onde o *pin* vai ficar inserido.
- ♦ **Informação da restauração:** Na nossa aplicação web é possível ver a informação dos produtos que a restauração oferece por imagens, como por exemplo o preço dos mesmos. Seria interessante que como administrador ao alterar algum desses produtos os preços nas imagens também fossem atualizados.
- ♦ **Gestão de permissões:** Uma ideia que seria interessante para o futuro era fazer com que o gerente conseguisse fazer uma gestão de permissões para cada cargo que existe ou que seja criado. Desta forma o gerente tinha a capacidade de controlar o acesso das diferentes páginas para cada cargo.

18.5 Síntese das Experiências

A unidade curricular de Projeto Temático em Aplicações Web tem como objetivo a aplicação dos conhecimentos obtidos juntamente com a unidade curricular que compõem o módulo temático, permitindo aos alunos a experiência de trabalho de equipa com vista no cumprimento de um conjunto de objetivos.

O projeto fluiu muito bem e sem conflitos, ainda assim, nos momentos mais atribulados, existiu sempre comunicação e entajuda no grupo, permitindo um trabalho de equipa eficiente.

18.6 Conclusão

Com a realização deste projeto, foram postos em prática os conhecimentos adquiridos das unidades curriculares lecionadas no passado como unidades curriculares que estão associadas ao Projeto Temático, Programação e Tecnologias Web, lecionada pelo docente Ciro Martins.

Durante a realização do projeto houve algumas mudanças nas ideias originais do grupo e foram feitas alterações durante o processo de desenvolvimento para melhorar a qualidade do que foi proposto desenvolver inicialmente.

O grupo superou a maior parte das dificuldades com que se deparou, como por exemplo a conexão inicial com a base de dados e implementar todas as ideias que foram propostas.

A elaboração deste projeto proporcionou um excelente desafio que maturou a amizade e a ética de trabalho já existente dos diversos elementos do grupo.

19. Bibliografia

- [1] «PHP QR Code - QR code generator, an LGPL PHP library». Disponível em: <https://phpqrcode.sourceforge.net/>.
- [2] P. G. D. Group, «PostgreSQL», *PostgreSQL*, 28 de maio de 2024. Disponível em: <https://www.postgresql.org/>.
- [3] «Stack Overflow - Where Developers Learn, Share, & Build Careers». Disponível em: <https://stackoverflow.com/>.
- [4] «TCPDF». Disponível em: <https://tcpdf.org/>.
- [5] «Testes de Sistema - Tipos, Processo, Ferramentas & Mais!». Disponível em: <https://www.zaptest.com/pt-pt/o-que-sao-testes-de-sistema-um-mergulho-profundo-nas-abordagens-tipos-ferramentas-dicas-e-truques-e-muito-mais>.
- [6] «Visual Paradigm». Disponível em: <https://www.visual-paradigm.com/>.
- [7] «Visual Studio Code - Code Editing. Redefined». Disponível em: <https://code.visualstudio.com/>.
- [8] «W3Schools Online Web Tutorials». Disponível em: <https://www.w3schools.com/>.
- [9] «XAMPP». Disponível em: <https://www.apachefriends.org/>.
- [10] «Zoom», *Zoom*. Disponível em: <https://zoom.us>.
- [11] «Zoomarine Algarve - Parque Temático, Zoológico e Aquático». Disponível em: <https://www.zoomarine.pt/pt/>.
- [12] «Parque Aquático Amarante». Disponível em: <https://parqueaquaticoamarante.com/>.
- [13] «Página Principal - Vagasplash». Disponível em: <https://www.vagasplash.com/>.
- [14] «Norpark • Parque Aquático». Disponível em: <https://norpark.pt/>.
- [15] Ebsss, «Natur Waterpark». Disponível em: <https://naturwaterpark.pt/inicio>.
- [16] «Microsoft Teams». Disponível em: <https://www.microsoft.com/pt-pt/microsoft-teams/log-in>.
- [17] «JavaScript Documentation». Disponível em: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- [18] «How to Use Git and GitHub – Introduction for Beginners», *freeCodeCamp.org*, 26 de setembro de 2022. Disponível em: <https://www.freecodecamp.org/news/introduction-to-git-and-github/>.
- [19] «Git». Disponível em: <https://git-scm.com/>.
- [20] «Trello». Disponível em: <https://trello.com/pt-BR>.
- [21] «FileZilla». Disponível em: <https://filezilla-project.org/>.
- [22] «Figma: The Collaborative Interface Design Tool». Disponível em: <https://www.figma.com/>.
- [23] «Discord». Disponível em: <https://discord.com/>. [Acedido: 29 de maio de 2024]
- [24] M. O. contributors Jacob Thornton, and Bootstrap, «Bootstrap». Disponível em: <https://getbootstrap.com/>.
- [25] «Balsamiq: Fast, focused wireframing for teams and individuals | Balsamiq». Disponível em: <https://balsamiq.com/>.
- [26] «Arduino». Disponível em: <https://www.arduino.cc/en/software>.

ANEXO A

Teste de usabilidade - Levantamento das grelhas de observação

Tarefa		Execução			Erros	Dificuldades	Dúvidas	Observações	Duração
#	Descrição	Bastante Dificuldade	Alguma Dificuldade	Sem Dificuldade					
Tarefas como funcionário									03:10.37
1	Verificar o horário de funcionamento do parque			X					00:09.63
2	Perguntar ao assistente de IA sobre o horário do parque			X					00:08.34
3	Realizar a compra de algo na loja		X				1	Dúvida de como voltar para a página inicial do parque, sem ser a interface da loja	01:45.16
4	Ver histórico de compras		X			1		Dificuldade em perceber o histórico de compras devido a má organização	00:37:34
5	Visitar o mapa do parque			X					00:11.68
6	Alugar um Bungalow por 4 dias			X	1			Foi detetado um erro, pois o utilizador conseguiu reservar um Bungalow para um dia anterior ao dia do teste	00:55.31

Tarefa		Execução			Erros	Dificuldades	Dúvidas	Observações	Duração
#	Descrição	Bastante Dificuldade	Alguma Dificuldade	Sem Dificuldade					
Tarefas como Gerente									01:30.37
1	Fazer login como Gerente			X					00:16.80
2	Editar uma zona do parque			X					00:12.00
3	Desligar a iluminação do parque		X			1		Dificuldade ao perceber onde se encontrava a Dashboard de iluminação do parque	00:28.29
4	Editar algo na área de restauração		X		1			Erro ao adicionar imagem	00:19.20
5	Fazer download do PDF das estatísticas			X					00:14.38
6	Eliminar categoria	X				1		Impossibilitado de eliminar categoria devido a existir vários objetos relacionados com essa categoria	1:02.23

Tarefa		Execução			Erros	Dificuldades	Dúvidas	Observações	Duração
#	Descrição	Bastante Dificuldade	Alguma Dificuldade	Sem Dificuldade					
Tarefas como funcionário									03:68.77
1	Verificar o horário de funcionamento do parque			X					00:12.80
2	Perguntar ao assistente de IA sobre o horário do parque			X					00:06.68
3	Realizar a compra de algo na loja			X					01:10.42
4	Ver histórico de compras		X			1		Dificuldade em ver se o bilhete se encontrava válido ou não	00:35.32
5	Visitar o mapa do parque			X					00:20.23
6	Alugar um Bungalow por 4 dias			X			1	O utilizador perguntou como é que se sabia a disponibilidade para o dia que ele queria	1:20.32

Tarefa		Execução			Erros	Dificuldades	Dúvidas	Observações	Duração
#	Descrição	Bastante Dificuldade	Alguma Dificuldade	Sem Dificuldade					
Tarefas como Gerente									02:32.32
1	Fazer login como Gerente			X					00:17.88
2	Editar uma zona do parque			X					00:15.30
3	Desligar a iluminação do parque			X					00:20.01
4	Editar algo na área de restauração			X					00:20.63
5	Fazer download do PDF das estatísticas			X					00:13.20
6	Eliminar categoria			X					00:58.30

ANEXO B

Teste de usabilidade - Levantamento do questionário de satisfação

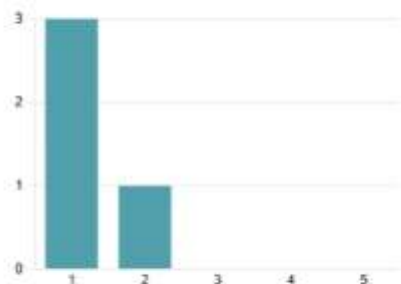
1. Gostaria de utilizar este sistema com frequência (0 ponto)

4.25
Classificação Média



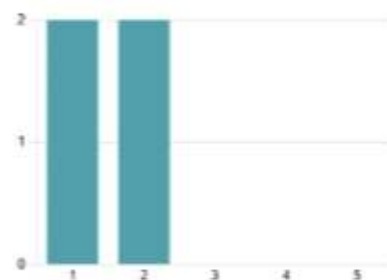
2. O sistema é desnecessariamente complexo (0 ponto)

1.25
Classificação Média



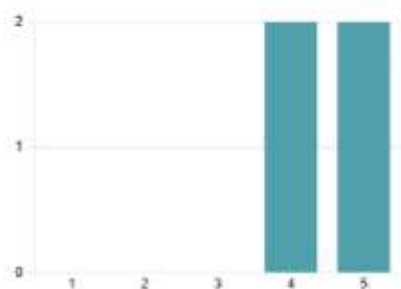
3. É necessário que eu precise de ajuda de uma pessoa com conhecimentos técnicos para utilizar o sistema (0 ponto)

1.50
Classificação Média



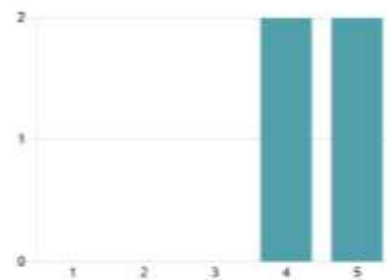
3. Achei o sistema fácil de utilizar (0 ponto)

4.50
Classificação Média

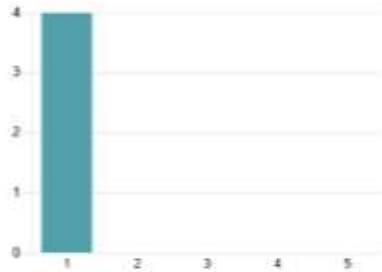


4. Acredito que as várias funções do sistema estão bem integradas (0 ponto)

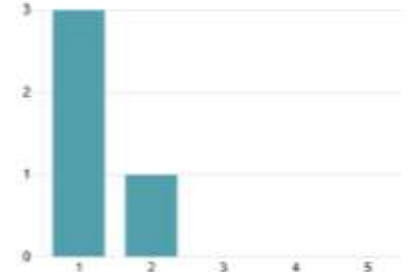
4.50
Classificação Média



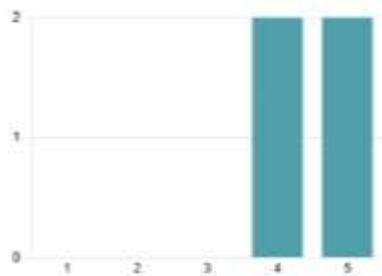
6. Acho que o sistema apresenta muita inconsistência (0 ponto)

1.00
Classificação Média

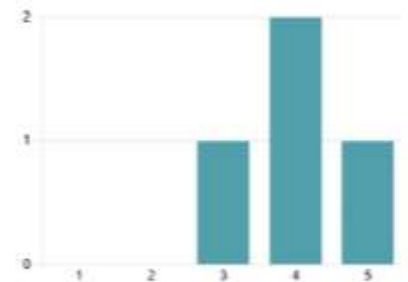
1. Achei que o sistema era confuso de utilizar (0 ponto)

1.25
Classificação Média

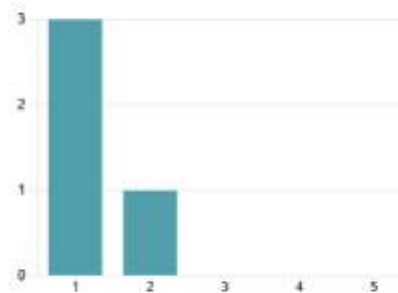
7. Penso que a maioria das pessoas aprenderia rapidamente como utilizar o sistema (0 ponto)

4.50
Classificação Média

1. Senti-me confiante ao utilizar o sistema (0 ponto)

4.00
Classificação Média

antes de conseguir utilizar o sistema (0 ponto)

1.25
Classificação Média

ANEXO D

Script Base de Dados


```
CREATE TABLE IF NOT EXISTS alertas (  
    id_alerta SERIAL PRIMARY KEY,  
    data TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    mensagem TEXT,  
    lido BOOLEAN DEFAULT FALSE  
);  
  
CREATE TABLE IF NOT EXISTS aluguer (  
    id_aluguer SERIAL PRIMARY KEY,  
    quantidade INTEGER,  
    hora_inicio TIMESTAMP,  
    hora_fim TIMESTAMP,  
    id_produto INTEGER NOT NULL,  
    id_compra INTEGER NOT NULL,  
    status BOOLEAN DEFAULT FALSE,  
    recolha TIMESTAMP,  
    id_funcionario INTEGER  
);  
  
CREATE TABLE IF NOT EXISTS bilhete (  
    id_bilhete SERIAL PRIMARY KEY,  
    data_inicio DATE,  
    data_fim DATE,  
    tipo VARCHAR(512),  
    codigo INTEGER,  
    status BOOLEAN,  
    preco REAL,  
    id_funcionario INTEGER NOT NULL,  
    id_produto INTEGER NOT NULL,  
    data_compra TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    id_utilizador INTEGER NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS cargo (  
    id_cargo SERIAL PRIMARY KEY,  
    descricao VARCHAR(512)  
);  
  
CREATE TABLE IF NOT EXISTS categoria (  
    id_categoria SERIAL PRIMARY KEY,  
    nome VARCHAR(512),  
    loja_online BOOLEAN DEFAULT FALSE,  
    visibilidade BOOLEAN DEFAULT TRUE NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS compra (  
    id_compra SERIAL PRIMARY KEY,  
    morada_entrega VARCHAR(512),  
    id_utilizador INTEGER NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS compra_detalhe (  
    data_hora DATE,  
    quantidade INTEGER,  
    preco_total DOUBLE PRECISION,
```

```
id_produto INTEGER NOT NULL,  
id_compra INTEGER NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS evento (  
  id_evento SERIAL PRIMARY KEY,  
  data_inicio DATE,  
  data_fim DATE,  
  tipo VARCHAR(512),  
  descricao VARCHAR(512),  
  num_convitados INTEGER,  
  id_utilizador INTEGER NOT NULL,  
  id_zona INTEGER NOT NULL,  
  aprovado BOOLEAN DEFAULT FALSE  
);  
  
CREATE TABLE IF NOT EXISTS feedback (  
  id_feedback SERIAL PRIMARY KEY,  
  data DATE,  
  descricao VARCHAR(512),  
  avaliacao REAL,  
  id_utilizador INTEGER NOT NULL,  
  status BOOLEAN DEFAULT FALSE NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS funcionario (  
  id_funcionario SERIAL PRIMARY KEY,  
  nome VARCHAR(512),  
  password VARCHAR(512),  
  email TEXT,  
  img TEXT DEFAULT 'https://www.shutterstock.com/image-  
vector/default-avatar-profile-icon-social-600nw-1677509740.jpg'  
);  
  
CREATE TABLE IF NOT EXISTS horario (  
  dia INTEGER,  
  hora_a TEXT,  
  hora_f TEXT,  
  modalidade TEXT,  
  id SERIAL PRIMARY KEY  
);  
  
CREATE TABLE IF NOT EXISTS luzes (  
  id_luz SERIAL PRIMARY KEY,  
  status BOOLEAN DEFAULT FALSE NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS noticias (  
  id_noticia SERIAL PRIMARY KEY,  
  imagem_capa TEXT DEFAULT  
'https://media.discordapp.net/attachments/1215368103626805388/121537121  
1802939443/codewave_logo.jpg?ex=6649a165&is=66484fe5&hm=8dc11607f4aeff4  
fc761edcaed53ee748831bb78aa62614edcdd1996ed870807&=&format=webp&width=1  
068&height=671',  
  titulo_noticia VARCHAR(512),  
  funcionario_criador VARCHAR(512),  
  data_hora TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
  descricao_noticia TEXT
```

```
);

CREATE TABLE IF NOT EXISTS pagamento (
    id_pagamento SERIAL PRIMARY KEY,
    preco_total REAL,
    tipo_pagamento VARCHAR(512),
    id_compra INTEGER NOT NULL
);

CREATE TABLE IF NOT EXISTS produto (
    id_produto SERIAL PRIMARY KEY,
    nome VARCHAR(512),
    preco REAL,
    capacidade INTEGER,
    iva REAL,
    id_categoria INTEGER NOT NULL,
    id_zona INTEGER,
    imagem TEXT
);

CREATE TABLE IF NOT EXISTS progressao (
    ano_inicio DATE,
    ano_fim DATE,
    id_funcionario INTEGER NOT NULL,
    id_cargo INTEGER NOT NULL,
    local TEXT,
    idade TEXT,
    salario DOUBLE PRECISION DEFAULT 900,
    PRIMARY KEY (id_funcionario, id_cargo)
);

CREATE TABLE IF NOT EXISTS utilizador (
    id_utilizador SERIAL PRIMARY KEY,
    nome VARCHAR(512),
    email VARCHAR(512),
    password VARCHAR(512),
    num_telemovei VARCHAR(512),
    newsletter BOOLEAN DEFAULT FALSE
);

CREATE TABLE IF NOT EXISTS zona (
    id_zona SERIAL PRIMARY KEY,
    nome VARCHAR(512)
);

ALTER TABLE aluguer ADD CONSTRAINT aluguer_compra_fk FOREIGN KEY
(id_compra) REFERENCES compra(id_compra);
ALTER TABLE aluguer ADD CONSTRAINT aluguer_produto_fk FOREIGN KEY
(id_produto) REFERENCES produto(id_produto);
ALTER TABLE bilhete ADD CONSTRAINT bilhete_funcionario_fk FOREIGN KEY
(id_funcionario) REFERENCES funcionario(id_funcionario);
ALTER TABLE bilhete ADD CONSTRAINT bilhete_produto_fk FOREIGN KEY
(id_produto) REFERENCES produto(id_produto);
ALTER TABLE progressao ADD CONSTRAINT cargo_progressao_fk FOREIGN KEY
(id_cargo) REFERENCES cargo(id_cargo);
ALTER TABLE compra_detalhe ADD CONSTRAINT compra_detalhe_compra_fk
FOREIGN KEY (id_compra) REFERENCES compra(id_compra);
```

```
ALTER TABLE compra_detalhe ADD CONSTRAINT compra_detalhe_produto_fk
FOREIGN KEY (id_produto) REFERENCES produto(id_produto);
ALTER TABLE compra ADD CONSTRAINT compra_utilizador_fk FOREIGN KEY
(id_utilizador) REFERENCES utilizador(id_utilizador);
ALTER TABLE evento ADD CONSTRAINT evento_utilizador_fk FOREIGN KEY
(id_utilizador) REFERENCES utilizador(id_utilizador);
ALTER TABLE evento ADD CONSTRAINT evento_zona_fk FOREIGN KEY (id_zona)
REFERENCES zona(id_zona);
ALTER TABLE feedback ADD CONSTRAINT feedback_utilizador_fk FOREIGN KEY
(id_utilizador) REFERENCES utilizador(id_utilizador);
ALTER TABLE progressao ADD CONSTRAINT funcionario_progressao_fk FOREIGN
KEY (id_funcionario) REFERENCES funcionario(id_funcionario);
ALTER TABLE pagamento ADD CONSTRAINT pagamento_compra_fk FOREIGN KEY
(id_compra) REFERENCES compra(id_compra);
ALTER TABLE produto ADD CONSTRAINT produto_categoria_fk FOREIGN KEY
(id_categoria) REFERENCES categoria(id_categoria);
ALTER TABLE produto ADD CONSTRAINT produto_zona_fk FOREIGN KEY
(id_zona)
```

```
CREATE OR REPLACE FUNCTION trg_novo_produto()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN
    INSERT INTO alertas (mensagem) VALUES ('Novo produto adicionado: '
|| NEW.nome);
    RETURN NEW;
END;
$$;

CREATE OR REPLACE FUNCTION adicionar_alerta_cargo()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN
    INSERT INTO alertas (mensagem) VALUES ('Novo cargo adicionado: ' ||
NEW.descricao);
    RETURN NEW;
END;
$$;

CREATE OR REPLACE FUNCTION trg_nova_categoria()
RETURNS TRIGGER
LANGUAGE plpgsql
AS $$
BEGIN
    INSERT INTO alertas (mensagem) VALUES ('Nova categoria adicionada:
' || NEW.nome);
    RETURN NEW;
END;
$$;
```

```
CREATE OR REPLACE FUNCTION trg_nova_zona()
  RETURNS TRIGGER
  LANGUAGE plpgsql
AS $$
BEGIN
  INSERT INTO alertas (mensagem) VALUES ('Nova zona adicionada: ' ||
NEW.nome);
  RETURN NEW;
END;
$$;

CREATE OR REPLACE FUNCTION trg_novo_aluguer()
  RETURNS TRIGGER
  LANGUAGE plpgsql
AS $$
DECLARE
  nome_produto_text TEXT;
BEGIN
  SELECT nome INTO nome_produto_text FROM produto WHERE id_produto =
NEW.id_produto;

  INSERT INTO alertas (mensagem) VALUES ('Novo aluguer do produto: '
|| nome_produto_text);

  RETURN NEW;
END;
$$;

CREATE OR REPLACE FUNCTION trg_novo_evento()
  RETURNS TRIGGER
  LANGUAGE plpgsql
AS $$
BEGIN
  INSERT INTO alertas (mensagem) VALUES ('Novo evento adicionado: '
|| NEW.tipo);
  RETURN NEW;
END;
$$;

CREATE OR REPLACE FUNCTION trg_novo_horario()
  RETURNS TRIGGER
  LANGUAGE plpgsql
AS $$
BEGIN
  INSERT INTO alertas (mensagem) VALUES ('Novo horário: ' ||
NEW.modalidad);
  RETURN NEW;
END;
$$;
```

ANEXO E

Atas

**PROJETO TEMÁTICO EM APLICAÇÕES WEB**
AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENCAS
NÚMERO: 1	ASS: LUCAS DUARTE
DATA: 07/03/2024	ASS: GABRIEL CRAVO
PIVÔ RESPONSÁVEL: LUCAS DUARTE	ASS: MIGUEL PIRRE
HORA INÍCIO: 16:00	ASS: DANIEL SILVA
HORA TÉRMINO: 16:25	ASS: BRUNO MIGUEIS
PRÓXIMA REUNIÃO: 14/03/2024	ASS:

**1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES****2. ATA DA REUNIÃO**

Aprovação da ideia do projeto;

Definição de tarefas iniciais para o projeto;

Meios de comunicação entre os elementos do grupo;

3. COMENTÁRIOS / NOTAS

A reunião correu dentro da normalidade.

(todos os campos apresentados a cinza devem vir preenchidos para a reunião)



ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO DE ÁGUEDA

UNIVERSIDADE DE AVEIRO

PROJETO TEMÁTICO EM APLICAÇÕES WEB
AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENCAS
NÚMERO: 2	ASS: BRUNO MIGUEIS
DATA: 14/03/2024	ASS: LUCAS DUARTE
PIVÔ RESPONSÁVEL: BRUNO MIGUEIS	ASS: GABRIEL CRAVO
HORA INÍCIO: 16:00	ASS: DANIEL SILVA
HORA TÉRMINO: 16:45	ASS: MIGUEL PIRRÉ
PRÓXIMA REUNIÃO: 21/03/2024	ASS:

1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES

Apresentação de diagramas de casos de uso e classes;

Apresentação do relatório de especificação de requisitos;

Apresentação do diagrama de Gantt;

Discussão sobre os próximos passos a executar.

2. ATA DA REUNIÃO

Discutiram-se os diagramas e os requisitos já levantados, juntamente com o diagrama de Gantt.

Foram apresentadas melhorias e/ou críticas construtivas acerca dos mesmos por parte do orientador.

Apontaram-se os próximos passos a executar no projeto.

3. COMENTÁRIOS / NOTAS

A reunião correu como esperado, dentro da normalidade com a duração de 45 minutos.

{todos os campos apresentados a cinza devem vir preenchidos para a reunião}

**PROJETO TEMÁTICO EM APLICAÇÕES WEB**
AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENCAS
NÚMERO: 3	ASS: BRUNO MIGUEIS
DATA: 21/03/2024	ASS: LUCAS DUARTE
PIVÔ RESPONSÁVEL: DANIEL SILVA	ASS: MIGUEL PIRRE
HORA INÍCIO: 16:00	ASS: DANIEL SILVA
HORA TÉRMINO: 16:40	ASS:
PRÓXIMA REUNIÃO: 28/03/2024	ASS:

1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES

Apresentação da correção dos diagramas de casos de uso e classes;

Apresentação do protótipo de baixa fidelidade;

Discussão sobre os próximos passos a executar.

2. ATA DA REUNIÃO

Durante a reunião foram discutidos os novos diagramas apresentados e percebeu-se onde havia espaço para melhorias e reflexão.

Foi discutido também o protótipo de baixa fidelidade e respetivas críticas construtivas.

Foram propostos os próximos passos a seguir.

3. COMENTÁRIOS / NOTAS

A reunião correu como esperado, dentro da normalidade e cordialidade de ambas as partes.

PROJETO TEMÁTICO EM APLICAÇÕES WEB
AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENÇAS
NÚMERO: 4	ASS: MIGUEL PIRRÉ
DATA: 28/03/2024	ASS: GABRIEL CRAVO
PIVÔ RESPONSÁVEL: MIGUEL PIRRÉ	ASS: ARTIOM MASLIUCOV
HORA INÍCIO: 16:00	ASS: BRUNO MIGUEIS
HORA TÉRMINO: 16:43	ASS: LUCAS DUARTE
PRÓXIMA REUNIÃO: 04/04/2024	

1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES

Apresentação de diagramas de casos de uso e classes com as alterações propostas na reunião passada;

Apresentação do website;

Discussão sobre os próximos passos a executar.

2. ATA DA REUNIÃO

A reunião correu como esperado, todos os pontos foram abordados e foram planeados os próximos passos do projeto.

3. COMENTÁRIOS / NOTAS

(todos os campos apresentados a [cinza](#) devem vir preenchidos para a reunião)

**PROJETO TEMÁTICO EM APLICAÇÕES WEB**
AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENCAS
NÚMERO: 5	ASS: LUCAS DUARTE
DATA: 04/04/2024	ASS: GABRIEL CRAVO
PIVÔ RESPONSÁVEL: GABRIEL CRAVO	ASS: MIGUEL PIRRE
HORA INÍCIO: 16:00	ASS: DANIEL SILVA
HORA TÉRMINO: 16.32	ASS: BRUNO MIGUEIS
PRÓXIMA REUNIÃO: 11/04/2024	ASS:

1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES

Esclarecimento sobre os modelos da base de dados;

Definição das próximas tarefas para o projeto;

2. ATA DA REUNIÃO

Houve o esclarecimento de dúvidas sobre os modelos da base de dados.

Ponto de situação do projeto.

Definição de ordem de trabalhos até à próxima reunião.

3. COMENTÁRIOS / NOTAS

Reunião correu como previsto.

(todos os campos apresentados a cinza devem vir preenchidos para a reunião)



PROJETO TEMÁTICO EM APLICAÇÕES WEB

AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENCAS
NÚMERO: 6	ASS: ARTIOM MASLIUCOV
DATA: 18/04/2024	ASS: LUCAS DUARTE
PIVÔ RESPONSÁVEL: ARTIOM MASLIUCOV	ASS: GABRIEL CRAVO
HORA INÍCIO: 16:00	ASS: DANIEL SILVA
HORA TÉRMINO: 16:40	ASS: BRUNO MIGUEIS
PRÓXIMA REUNIÃO: 25/04/2024	ASS: MIGUEL PIRRÉ

1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES

Demonstração do chat Ollama;

Apresentação da base de dados;

Discussão sobre os próximos passos a executar.

2. ATA DA REUNIÃO

A reunião correu como esperado, foram tiradas dúvidas sobre o projeto e foi discutido os próximos passos para o projeto.

3. COMENTÁRIOS / NOTAS

(todos os campos apresentados a cinza devem vir preenchidos para a reunião)

PROJETO TEMÁTICO EM APLICAÇÕES WEB
AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENCAS
NÚMERO: 7	ASS: LUCAS DUARTE
DATA: 25/04/2024	ASS: GABRIEL CRAVO
PIVÔ RESPONSÁVEL: LUCAS DUARTE	ASS: BRUNO MIGUEIS
HORA INÍCIO: 16:00	ASS: DANI SILVA
HORA TÉRMINO: 16:37	ASS: MIGUEL PIRRE
PRÓXIMA REUNIÃO: 02/05/2024	ASS:

1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES

Questões relativas ao acesso á base de dados com o túnel;

Demonstração do site a comunicar com a base de dados;

Apresentação do Diagrama de Gantt;

Discussão sobre os próximos passos a executar;

2. ATA DA REUNIÃO

A reunião ocorreu dentro da normalidade.

Foram esclarecidas as dúvidas, e foi mostrado o trabalho semanal.

3. COMENTÁRIOS / NOTAS

(todos os campos apresentados a [cinza](#) *devem* vir preenchidos para a reunião)



ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO DE ÁGUEDA

UNIVERSIDADE DE AVEIRO

PROJETO TEMÁTICO EM APLICAÇÕES WEB
AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENCAS
NÚMERO: 8	ASS: BRUNO MIGUEIS
DATA: 09/05/2024	ASS: LUCAS DUARTE
PIVÔ RESPONSÁVEL: BRUNO MIGUEIS	ASS: GABRIEL CRAVO
HORA INÍCIO: 16:20	ASS: DANIEL SILVA
HORA TÉRMINO: 17:05	ASS: ARTIOM MASLIUCOV
PRÓXIMA REUNIÃO: 16/05/2024	ASS:

1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES

Dúvidas sobre a base de dados da ESTGA;

Demonstração de dashboards de administrador do site;

Demonstração da loja online;

Demonstração do mapa 3d interativo;

Demonstração do site no geral;

Discussão dos próximos passos e tarefas a executar.

2. ATA DA REUNIÃO

Foi demonstrado o site em execução, as dashboards de administradores, a página dos bilhetes, horários e da loja online.

Foi testado o acesso à base de dados da ESTGA.

Foram discutidos os próximos passos a executar e as tarefas necessárias a implementar.

3. COMENTÁRIOS / NOTAS

A reunião correu dentro da normalidade e serviu para esclarecer algumas dúvidas.

(todos os campos apresentados a cinza devem vir preenchidos para a reunião)



PROJETO TEMÁTICO EM APLICAÇÕES WEB

AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENÇAS
NÚMERO: 9	ASS: DANIEL SILVA
DATA: 16/05/2024	ASS: LUCAS DUARTE
PIVÔ RESPONSÁVEL: DANIEL SILVA	ASS: GABRIEL CRAVO
HORA INÍCIO: 16:00	ASS: BRUNO MIGUEIS
HORA TÉRMINO: 16:32	ASS: MIGUEL PIRRÉ
PRÓXIMA REUNIÃO: 23/05/2024	ASS: ARTIOM MASLIUCOV

1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES

Demonstração dos progressos feitos na aplicação web durante toda a semana.

Discussão dos próximos passos e tarefas a executar.

2. ATA DA REUNIÃO

Nesta reunião foram demonstrados o progresso e a evolução na elaboração do website em execução, durante a semana. Foi proposto pelo professor orientador o abrandamento da implementação de novas funcionalidades e que o foco fosse “limar” o que estava feito.

3. COMENTÁRIOS / NOTAS

A reunião correu dentro da normalidade e serviu para ficar explícito o caminho a seguir daqui para a frente.



PROJETO TEMÁTICO EM APLICAÇÕES WEB

AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENÇAS
NÚMERO: 10	ASS: LUCAS DUARTE
DATA: 23/05/2024	ASS: GABRIEL CRAVO
PIVÔ RESPONSÁVEL: MIGUEL PIRRÉ	ASS: BRUNO MIGUEIS
HORA INÍCIO: 16:00	ASS: DANI SILVA
HORA TÉRMINO: 16:49	ASS: MIGUEL PIRRÉ
PRÓXIMA REUNIÃO: 30/05/2024	ASS:

1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES

Demonstração da utilização do website que se encontra nas partes finais.

Apresentar os avanços do relatório ao professor.

2. ATA DA REUNIÃO

A reunião decorreu como esperado.

3. COMENTÁRIOS / NOTAS

PROJETO TEMÁTICO EM APLICAÇÕES WEB
AGENDA E ATA DE REUNIÃO (GRUPO 1)

REUNIÃO	PRESENÇAS
NÚMERO: 11	ASS: LUCAS DUARTE
DATA: 30/05/2024	ASS: GABRIEL CRAVO
PIVÔ RESPONSÁVEL: GABRIEL CRAVO	ASS: MIGUEL PIRRÉ
HORA INÍCIO: 16:00	ASS: DANIEL SILVA
HORA TÉRMINO: 16:17	ASS: BRUNO MIGUEIS
PRÓXIMA REUNIÃO:	ASS: ARTIOM MASLIUCOV

1. ORDEM DE TRABALHOS DA REUNIÃO / QUESTÕES

Exposição de dúvidas sobre o relatório.

Esclarecimento de dúvidas sobre as expectativas da apresentação final.

2. ATA DA REUNIÃO

Demonstração do relatório.

Esclarecimento de dúvidas sobre o relatório e a estrutura da futura apresentação.

3. COMENTÁRIOS / NOTAS

Reunião correu como esperado

(todos os campos apresentados a cinza devem vir preenchidos para a reunião)