



**University of Minho**  
School of Engineering

## Processo e Metodologias de Software – LEGSI, 2º Ano, 2024/25

### Projeto PMS – Momento 4.4

PL5 - Grupo 2

*3ª iteração do projeto “PMS-TUB”*

			
Afonso Martim Carvalho Leite a108552 <a href="mailto:a108552@alunos.uminho.pt">(a108552@alunos.uminho.pt)</a>	Inês Margarida Pinheiro de Melo a108628 <a href="mailto:a108628@alunos.uminho.pt">(a108628@alunos.uminho.pt)</a>	João Filipe Ferreira Ribeiro a108545 <a href="mailto:a108545@alunos.uminho.pt">(a108545@alunos.uminho.pt)</a>	Pedro Afonso Ribeiro Lopes a108734 <a href="mailto:a108734@alunos.uminho.pt">(a108734@alunos.uminho.pt)</a>

## Índice

<b>1. Project Defined Process</b>	<b>3</b>
1. Introdução	3
1.1 Propósito	3
1.2 Âmbito	3
1.3 Definições, Acrônimos e Abreviações	3
1.4 Referências	3
1.5 Visão Geral	3
2. Visão geral do caso de desenvolvimento	4
2.1 Modelo de ciclo de vida	4
2.2 Amostra dos planos de iteração	5
2.2.1 Fase Inception	5
2.2.2 Fase Elaboration	7
2.2.3 Fase Construction	9
2.2.4 Fase Transition	11
3. Fluxo de trabalho	14
4. Artefactos	15
5. Relatórios	22
6. Papéis	23
7. Diretrizes e Procedimentos Específicos do Projeto	26
<b>2. Risk List</b>	<b>27</b>
<b>3. Tools</b>	<b>32</b>
3.1 System STAR	32
3.2 ProjectLibre	32
3.3 Google Docs	32
3.4 Google Sheets	33
3.5 OpenUP	33
3.6 CMMi	33
3.7. Visual Studio Code	33
<b>4. Glossary</b>	<b>34</b>
<b>5. Vision</b>	<b>38</b>
1. Introdução	38
2. Posicionamento	38
2.1. Declaração do Problema	38
2.2. Declaração de Posicionamento do Produto	38
3. Descrição dos Stakeholders	39
3.1. Resumo dos Stakeholders	39
3.2. Ambiente do Utilizador	40
4. Visão Geral do Produto	41
4.1. Necessidades e características	41
5. Outros Requisitos do Produto	42

<b>6. System Wide Requirements</b>	<b>43</b>
1. Introdução	43
2. Requisitos Funcionais de todo o Sistema	43
3. Requisitos de Qualidade do Sistema	44
3.1. Usabilidade	44
3.2. Fiabilidade	44
3.3. Desempenho	45
3.4. Suporte	45
3.5. Segurança	45
4. Interfaces do Sistema	46
4.1 Interfaces do Utilizador	46
4.1.1 Aparência e iteratividade	46
4.1.2 Requisitos de Layout e Navegação	47
4.1.3 Consistência	47
4.1.4 Requisitos de Personalização do Utilizador	47
4.2 Interfaces para Sistemas Externos ou Dispositivos	48
4.2.1 Interface de software	48
4.2.2 Interfaces de Hardware	48
4.2.3 Interfaces de comunicação	48
5. Regras de Negócio	51
5.1. Privacidade	51
5.2. Restrições	51
5.3. Disponibilidade	51
6. Restrições do sistema	51
7. Conformidade do sistema	52
7.1 Requisitos de Licenciamento	52
7.2. Legalidades, Direitos de Autor e outros Avisos	52
7.3. Leis Aplicáveis	52
8. Documentação do Sistema	52
<b>Use Case Model</b>	<b>55</b>
<b>6. Use Case Specification</b>	<b>60</b>
<b>1. Architecture Notebook</b>	<b>124</b>
1. Objetivo	124
2. Objetivos e filosofia da arquitetura	124
3. Pressupostos e dependências	124
4. Requisitos significativos do ponto de vista da arquitetura	124
5. Decisões, restrições e justificações	124
6. Mecanismos de arquitetura	124
7. Principais abstrações	124
8. Camadas ou estrutura arquitetónica	125
9. Visão arquitetónica	125
<b>2. Test Case</b>	<b>134</b>
1. Criação de Conta	134
2. Login com Credenciais Válidas	134

<b>10. Proof of Concept</b>	<b>135</b>
1. Criar Conta	136
2. Autenticação	136
3. Página Inicial	137
4. Compra de bilhetes	139
5. Pagamento	140
6. Histórico de Viagens	142
7. Consulta de Horários	143
8. Localização dos Autocarros em Tempo Real	143
9. Consultar Bilhete	144
10. Perfil do passageiro	146
11. Editar Perfil	147

# 1. Project Defined Process

## 1. Introdução

### 1.1 Propósito

O "Project Defined Process" tem o propósito de auxiliar, orientar e apoiar os diferentes membros envolvidos no projeto.

Este, estabelece os processos que devem ser seguidos e adotados ao longo do desenvolvimento. As revisões realizadas e os requisitos definidos, fornecem uma base de apoio que contribui para uma visão mais clara do projeto. Além disso, este processo serve como referência para as etapas de aprovação.

### 1.2 Âmbito

Este projeto foi iniciado em resposta a uma solicitação dos Transportes Urbanos de Braga, com o objetivo de criar sistemas de transporte inteligentes e conectados, capazes de antecipar e responder, em tempo real, às necessidades dos passageiros

### 1.3 Definições, Acrônimos e Abreviações

PMS - TUB: Processos e Metodologias de Software - Transportes Urbanos de Braga

### 1.4 Referências

Enunciado do momento 2 e 4 que está disponíveis na Blackboard  
OpenUP  
CMMi

### 1.5 Visão Geral

Na descrição deste artefato, começamos por definir o seu objetivo e explicar os acrónimos associados ao projeto. Em seguida, detalhamos o ciclo de vida do projeto, especificando cada iteração e as datas previstas de entrega.

Após essa introdução, identificamos as diferentes fases de cada iteração, acompanhadas de uma breve explicação de cada uma delas. Junto a essas informações, serão listadas as tarefas a serem realizadas e as práticas correspondentes. Após um estudo detalhado, identificamos que algumas dessas tarefas e práticas são comuns a várias fases.

Em relação à fase de *Inception*, que será aprofundada na Unidade Curricular de Processos e Metodologias de Software, também destacamos os produtos de trabalho associados.

## 2. Visão geral do caso de desenvolvimento

### 2.1 Modelo de ciclo de vida

Este projeto conta com 3 etapas de controlo:

- A **1<sup>a</sup> Iteração** terá a entrega dos artefactos:

→ *Project Defined Process;*  
→ *Risk List;*  
→ *Tools;*  
→ *Glossary.*

Data Limite de Entrega: 15-11-2024

- A **2<sup>a</sup> Iteração** terá a entrega dos artefactos:

→ *Vision;*  
→ *System Wide Requirements;*  
→ *Use Case Model;*  
→ *Use Case Specification.*

Data Limite de Entrega: 06-12-2024

- A **3<sup>a</sup> Iteração** terá a entrega dos artefactos:

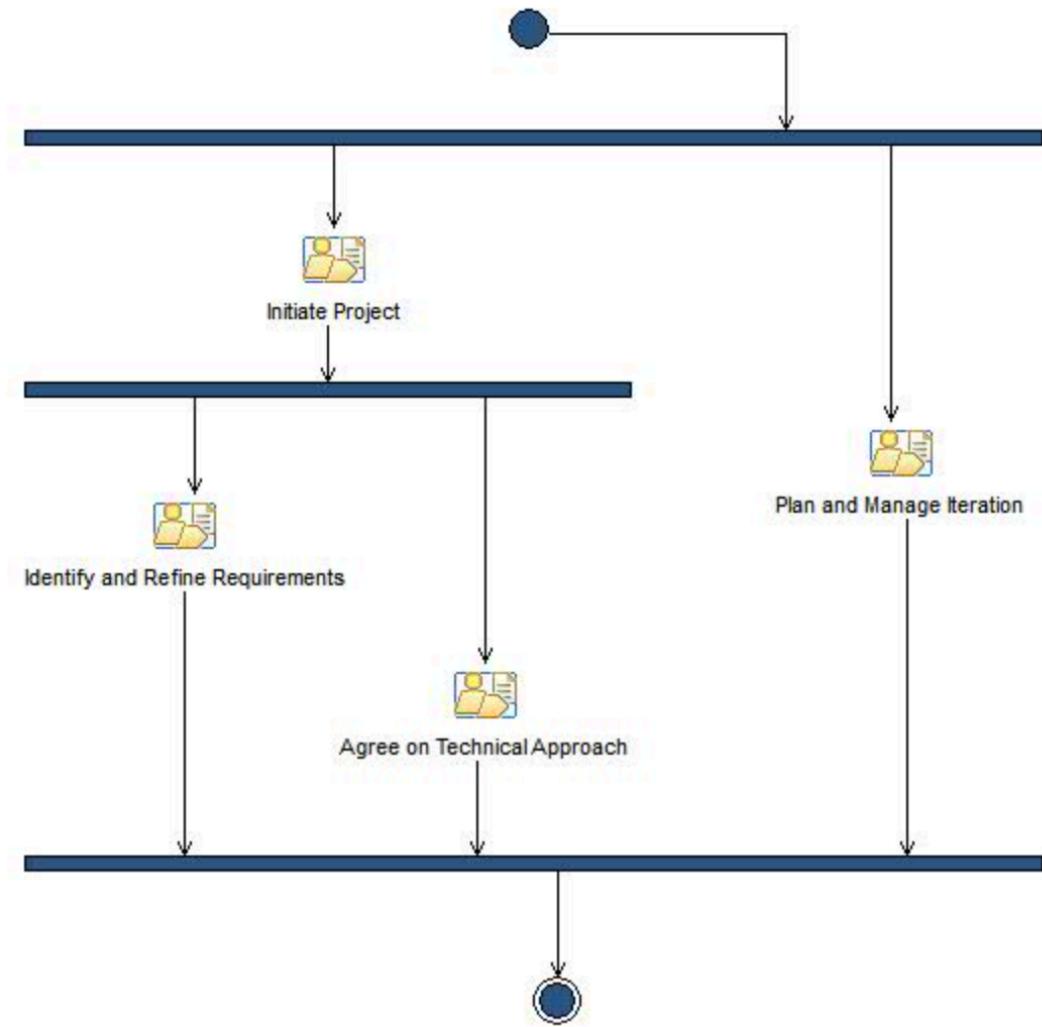
→ *Architecture Notebook;*  
→ *Test Case;*  
→ *Proof of Concept.*

Data Limite de Entrega: 20-12-2024

## 2.2 Amostra dos planos de iteração

### 2.2.1 Fase Inception

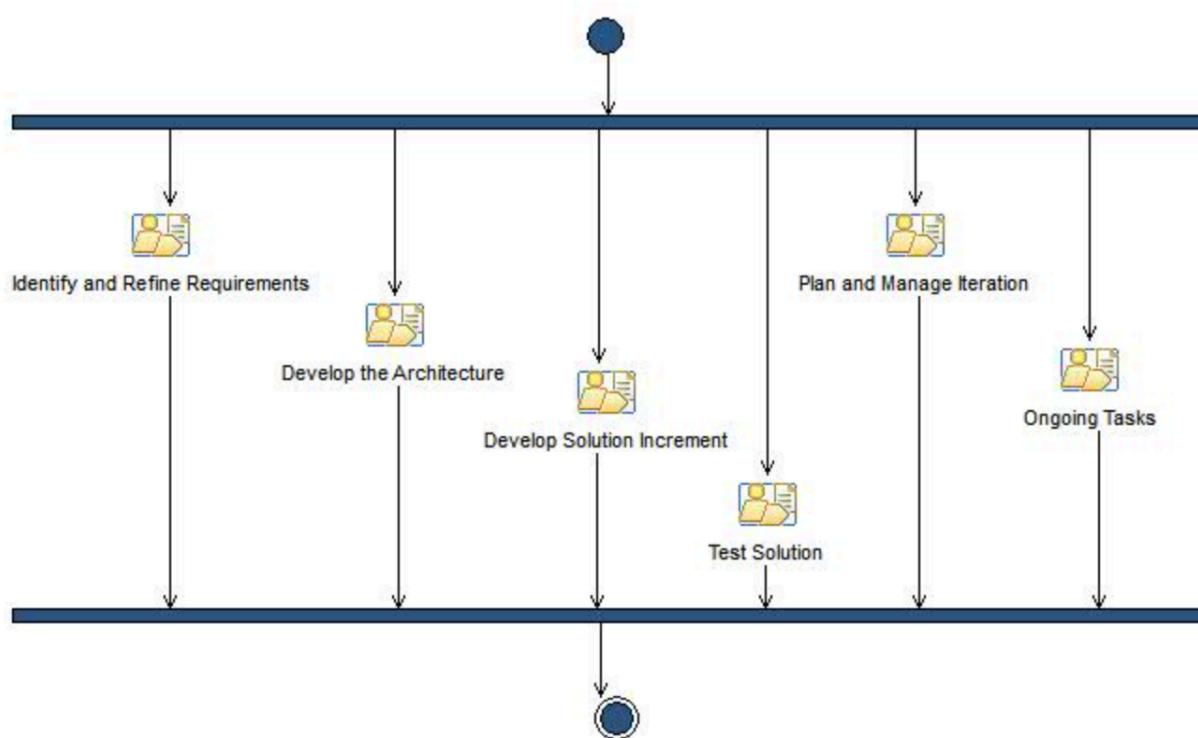
A fase de inception define a visão e o âmbito inicial do projeto, alinhando a equipa e os stakeholders nos objetivos, requisitos essenciais e expectativas. É uma fase de planeamento básico, onde se estabelece o propósito do projeto e os problemas que este visa resolver, criando uma base para o desenvolvimento do projeto.



Tarefas	Passos
Iniciar o Projeto	Desenvolver a Visão Técnica
	Planear o Projeto
Planear e Gerir a Iteração	Planear a Iteração
	Preparar o Ambiente:
	<ul style="list-style-type: none"> <li>● Adaptar o Processo</li> <li>● Configurar as Ferramentas</li> <li>● Verificar a instalação e configurar as Ferramentas</li> <li>● Implementar o processo</li> </ul>
	Gerir a Iteração
	Avaliar os Resultados
Identificar e Descrever os Requisitos	Detalhar os Requisitos
	Detalhar os cenários de Casos de Uso
	Detalhar os requisitos do Sistema
	Criar Casos de Teste
Acordar a Abordagem Técnica	Visualizar a Arquitetura

## 2.2.2 Fase Elaboration

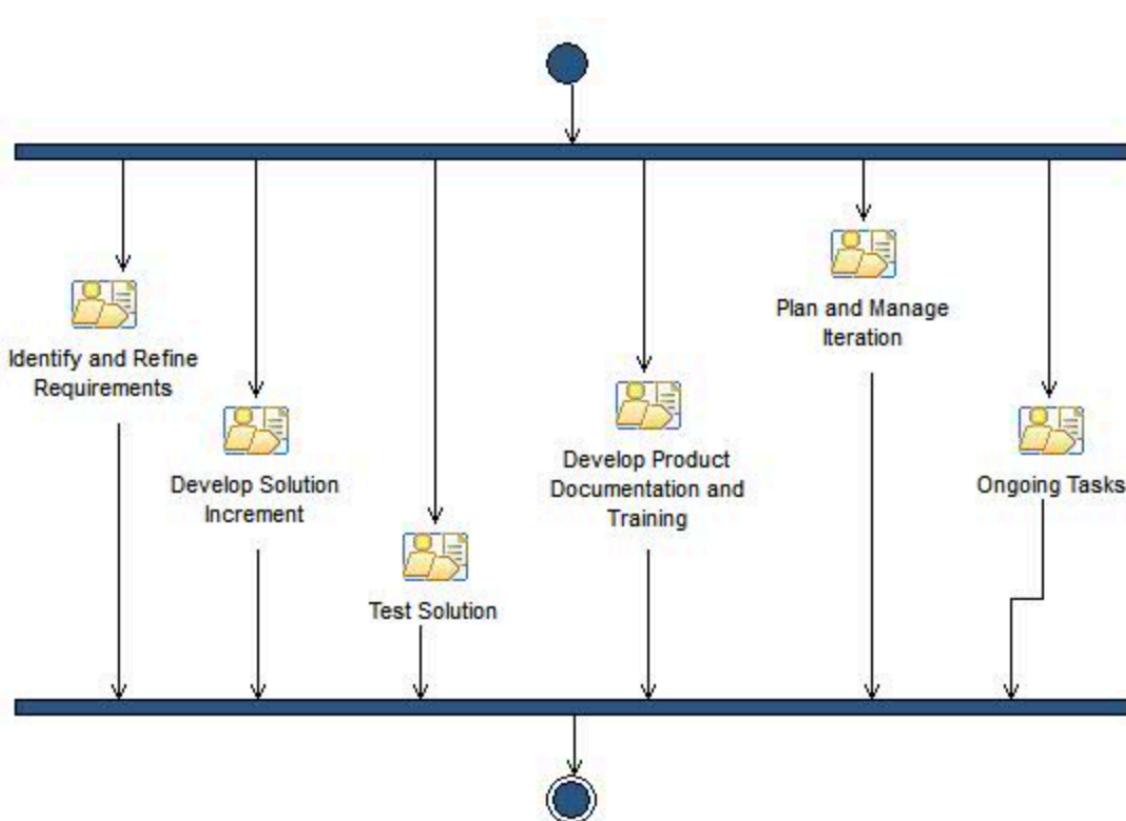
A fase de elaboration é a segunda etapa no ciclo de desenvolvimento de software, é nesta fase onde se detalha a arquitetura do sistema e se clarificam os requisitos principais. O objetivo principal desta fase é reduzir riscos e assegurar que a equipa tem uma compreensão clara do que será desenvolvido. Nesta fase refinam-se os requisitos essenciais, são geridos os riscos técnicos, e é realizado um planeamento mais detalhado para as próximas fases do projeto.



Tarefas	Passos
Planejar e Gerir a Iteração	Planejar a Iteração Preparar o Ambiente: <ul style="list-style-type: none"> <li>● Adaptar o Processo</li> <li>● Configurar as Ferramentas</li> <li>● Verificar a instalação e configurar as Ferramentas</li> <li>● Implementar o Processo</li> </ul> Gerir a Iteração Delinear o plano de Implementação Avaliar os Resultados
Identificar e Descrever os Requisitos	Detalhar os Requisitos Detalhar os cenários de Casos de Uso Detalhar os requisitos do Sistema Criar Casos de Teste
Desenvolver a Arquitetura	Aperfeiçoar a Arquitetura
Desenvolver o incremento da Solução	Desenhar a Solução Implementar Testes de Desenvolvimento Implementar a Solução Executar os Testes de Desenvolvimento Criar e Integrar a Construção
Testar a Solução	Implementar Testes Executar os Testes
Tarefas em Andamento	Solicitar alteração

### 2.2.3 Fase Construction

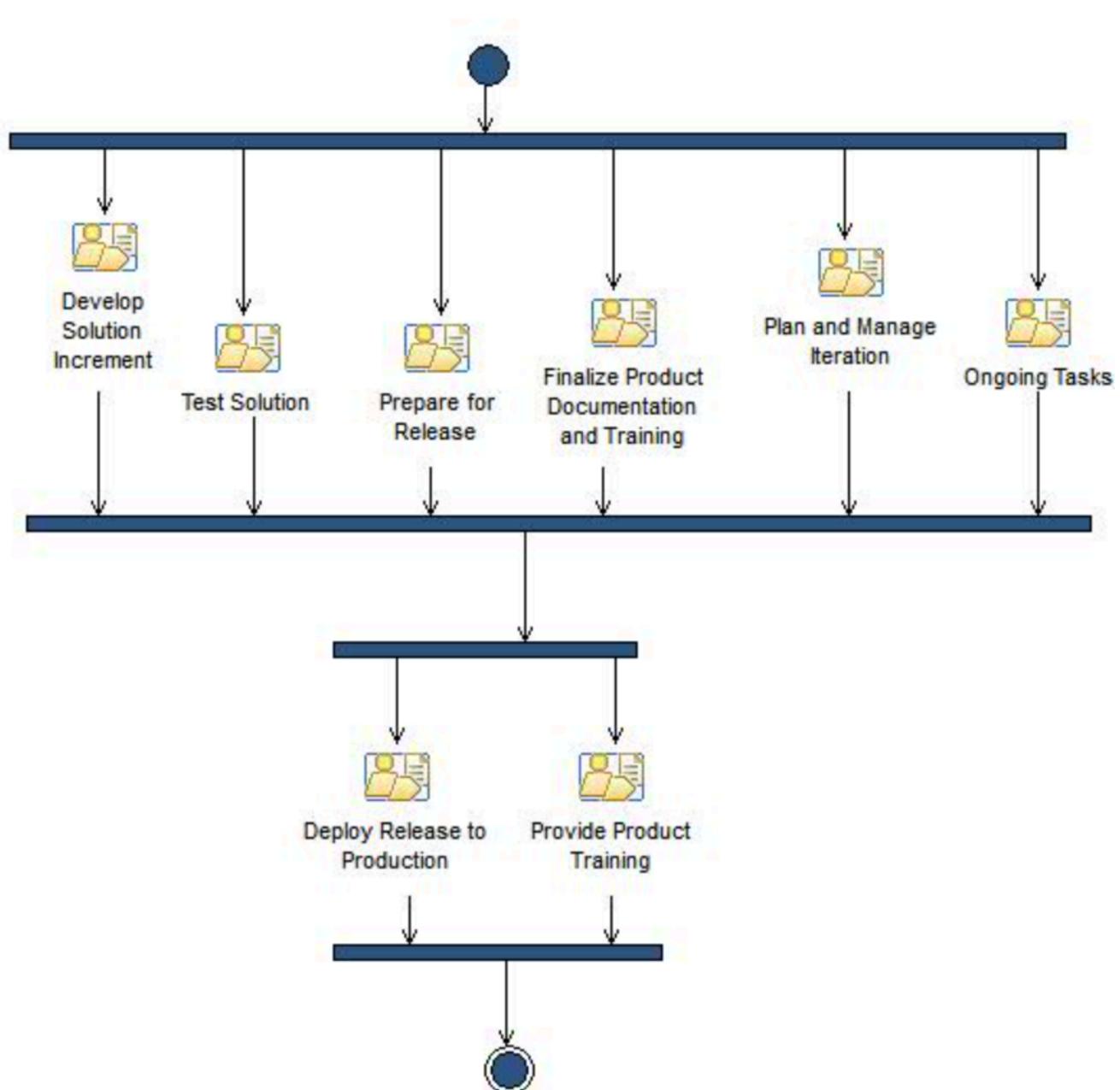
A fase de construction é a etapa em que o desenvolvimento do sistema é efetivamente realizado. Durante esta fase, a equipa de desenvolvimento cria, testa e implementa as funcionalidades de acordo com os requisitos estabelecidos nas fases anteriores. O foco principal nesta fase é construir o sistema de forma incremental, implementando as funcionalidades de maneira contínua até que o produto esteja pronto para poder ser entregue. Além disso, é realizada a verificação da qualidade do software, com testes contínuos, correção de falhas e aperfeiçoamento das funcionalidades.



Tarefas	Passos
Planejar e Gerir a Iteração	Planejar a Iteração Preparar o Ambiente: <ul style="list-style-type: none"> <li>● Adaptar o Processo</li> <li>● Configurar as Ferramentas</li> <li>● Verificar a Instalação e Configurar as Ferramentas</li> <li>● Implementar o Processo</li> </ul> Gerir a Iteração Avaliar os Resultados
Identificar e Descrever os Requisitos	Detalhar os Requisitos Detalhar os Cenários de Casos de Uso Detalhar os Requisitos do Sistema Criar Casos de Teste
Desenvolver o Incremento da Solução	Desenhar a Solução Implementar Testes de Desenvolvimento Implementar a Solução Executar os Testes de Desenvolvimento Criar e Integrar a Construção
Testar a Solução	Implementar Testes Executar os Testes
Tarefas em Andamento	Pedir Alteração
Desenvolver a Documentação e Treino do Produto	Desenvolver a Documentação do Produto Desenvolver a Documentação do Utilizador Desenvolver a Documentação de Suporte Desenvolver os Materiais de Treino

#### 2.2.4 Fase Transition

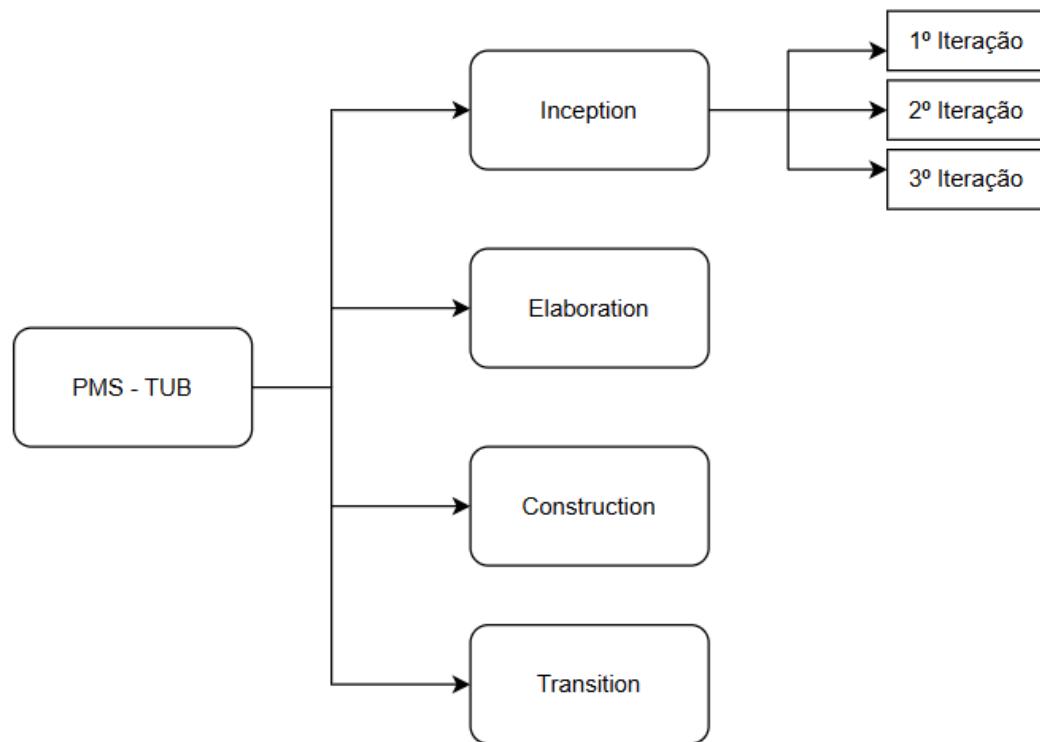
A fase de transition é a última etapa do ciclo de desenvolvimento, onde o sistema é entregue aos utilizadores finais. O objetivo principal é garantir que o software esteja pronto para ser utilizado em ambiente real.



Tarefas	Passos
Planejar e Gerir a Iteração	Planejar a Iteração Preparar o Ambiente: <ul style="list-style-type: none"> <li>● Adaptar o Processo</li> <li>● Configurar as Ferramentas</li> <li>● Verificar a Instalação e Configurar as Ferramentas</li> <li>● Implementar o Processo</li> </ul> Gerir a Iteração Avaliar os Resultados
Desenvolver o Incremento da Solução	Desenhar a Solução Implementar Testes de Desenvolvimento Implementar a Solução Executar os Testes de Desenvolvimento Criar e Integrar a Construção
Testar a Solução	Implementar Testes Executar os Testes
Finalizar a Documentação e o Treino do Produto	Desenvolver a Documentação do Produto Desenvolver a Documentação do Utilizador Desenvolver a Documentação de Suporte Desenvolver os Materiais de Treino
Preparar para o Lançamento	Rever a Conformidade com os Controlos de Lançamento Instalar e Validar a Infraestrutura Desenvolver o Plano de Recuperação Desenvolver Comunicações de Lançamento
Fornecer de Treino de Produtos	Entregar o Treino do Utilizador Final Entregar o Treino de Suporte
Tarefas em Andamento	Pedir Alteração
Implementar o Lançamento do Produto	Implementar o Lançamento

	Executar o Plano de Implementação
	Verificar o Sucesso da Implementação
	Executar o Plano de Recuperação (se necessário)
	Entregar Comunicações de Lançamento

### 3. Fluxo de trabalho



## 4. Artefactos

Tabela com os artefatos e respectivos passos da Fase **Inception**.

Task	Step	Artefactos
Iniciar o Projeto	Desenvolver a Visão Técnica	Glossary Vision
	Planejar o Projeto	Project Plan
Planejar e Gerir a Iteração	Planejar a Iteração	Iteration Plan Risk List Work Items List
	Preparar o Ambiente:	
	• Adaptar o Processo	Project Defined Process
	• Configurar as Ferramentas	Tools
	• Verificar a instalação e configurar as Ferramentas	
	• Implementar o processo	Project Defined Process
	Gerir a Iteração	Iteration Plan Risk List Work Items List
Identificar e Descrever os Requisitos	Avaliar os Resultados	Iteration Plan Work Items List
	Detalhar os Requisitos	Glossary System-Wide Requirements Use Case Use-Case Model Work Items List
Acordar a Abordagem Técnica	Visualizar a Arquitetura	Architecture Notebook

Tabela com os artefactos e respectivos passos da Fase **Elaboration**.

Task	Step	Artefactos
Planejar e Gerir a Iteração	Planejar a Iteração	Iteration Plan Risk List Work Items List
	Preparar o Ambiente:	
	• Adaptar o Processo	Project Defined Process
	• Configurar as Ferramentas	Tools
	• Verificar a instalação e configurar as Ferramentas	
	• Implementar o Processo	Project Defined Process
	Gerir a Iteração	Iteration Plan Risk List Work Items List
	Delinear o plano de Implementação	Implementation Plan
Identificar e Descrever os Requisitos	Avaliar os Resultados	Iteration Plan Work Items List
	Detalhar os Requisitos	Glossary System-Wide Requirements Use Case Use-Case Model Work Items List
	Detalhar os cenários de Casos de Uso	Glossary Use Case Use-Case Model
	Detalhar os requisitos do Sistema	Glossary System-Wide Requirements
Desenvolver a Arquitetura	Criar Casos de Teste	Test Case
	Aperfeiçoar a Arquitetura	Architecture Notebook
Desenvolver o Incremento da Solução	Desenhar a Solução	Design
	Implementar Testes de Desenvolvimento	Developer Test

	Implementar a Solução	Implementation
	Executar os Testes de Desenvolvimento	Test log
	Criar e Integrar a Construção	Build
Testar a Solução	Implementar Testes	Test Script
	Executar os Testes	Test log
Tarefas em Andamento	Solicitar alteração	Work Items List

Tabela com os artefatos e respectivos passos da Fase **Construction**.

Task	Step	Artefatos
Planejar e Gerir a Iteração	Planejar a Iteração	Iteration Plan Risk List Work Items List
	Preparar o Ambiente:	
	• Adaptar o Processo	Project Defined Process
	• Configurar as Ferramentas	Tools
	• Verificar a Instalação e Configurar as Ferramentas	
	• Implementar o Processo	Project Defined Process
	Gerir a Iteração	Iteration Plan Risk List Work Items List
Identificar e Descrever os Requisitos	Avaliar os Resultados	Iteration Plan Work Items List
	Detalhar os Requisitos	Glossary System-Wide Requirements Use Case Use-Case Model Work Items List
	Detalhar os Cenários de Casos de Uso	Glossary Use Case Use-Case Model
	Detalhar os Requisitos do Sistema	Glossary System-Wide Requirements
Desenvolver o Incremento da Solução	Criar Casos de Teste	Test Case
	Desenhar a Solução	Design
	Implementar Testes de Desenvolvimento	Developer Test
	Implementar a Solução	Implementation
	Executar os Testes de Desenvolvimento	Test log
Testar a Solução	Criar e Integrar a Construção	Build
	Implementar Testes	Script Test

	Executar os Testes	Test log
Tarefas em Andamento	Solicitar Alteração	Work Items List
Desenvolver a Documentação e Treino do Produto	Desenvolver a Documentação do Produto	Product Documentation
	Desenvolver a Documentação do Utilizador	User Documentation
	Desenvolver a Documentação de Suporte	User Documentation
	Desenvolver os Materiais de Treino	Training Materials

Tabela com os artefatos e respectivos passos da Fase da **Transition**.

Task	Step	Artefactos
Planejar e Gerir a Iteração	Planejar da Iteração	Iteration Plan Risk List Work Items List
	Preparar o Ambiente:	
	• Adaptar o Processo	Project Defined Process
	• Configurar as Ferramentas	Tools
	• Verificar a Instalação e Configurar as Ferramentas	
	• Implementar o Processo	Project Defined Process
	Gerir a Iteração	Iteration Plan Risk List Work Items List
Desenvolver o Incremento da Solução	Avaliar os Resultados	Iteration Plan Work Items List
	Desenhar a Solução	Design
	Implementar Testes de Desenvolvimento	Developer Test
	Implementar a Solução	Implementation
	Executar os Testes de Desenvolvimento	Test log
Testar a Solução	Criar e Integrar a Construção	Build
	Implementar Testes	Script Test
	Executar os Testes	Test log
Finalizar a Documentação e o Treino do Produto	Desenvolver a Documentação do Produto	Product Documentation
	Desenvolver a Documentação do Utilizador	User Documentation
	Desenvolver a Documentação do Suporte	User Documentation
	Desenvolver Materiais de Treino	Training Materials

Preparar para o Lançamento	Rever a conformidade dos Controlos de Lançamento	-----
	Instalar e Validar a Infraestrutura	Infrastructure
	Desenvolver o Plano de Recuperação	Backout Plan
	Desenvolver as Comunicações de Lançamento	Release Communications
Fornecer Treino de Produtos	Entregar o Treino do Utilizador Final	-----
	Entregar do Treino de Suporte	-----
Tarefas em Andamento	Pedir Alteração	Work Items List
Implementar o Lançamento do Produto	Implementar do Lançamento	Release
	Executar o Plano de Implementação	-----
	Verificar o Sucesso da Implementação	-----
	Executar o Plano de Recuperação (se necessário)	-----
	Entregar Comunicações de Lançamento	-----

## 5. Relatórios

Os relatórios são úteis para o acompanhamento e gestão do projeto, dado que desempenham um papel crucial na comunicação do progresso, qualidade, riscos e desempenho do projeto, permitindo que todos os envolvidos tomem decisões informadas e garantam o sucesso do projeto.

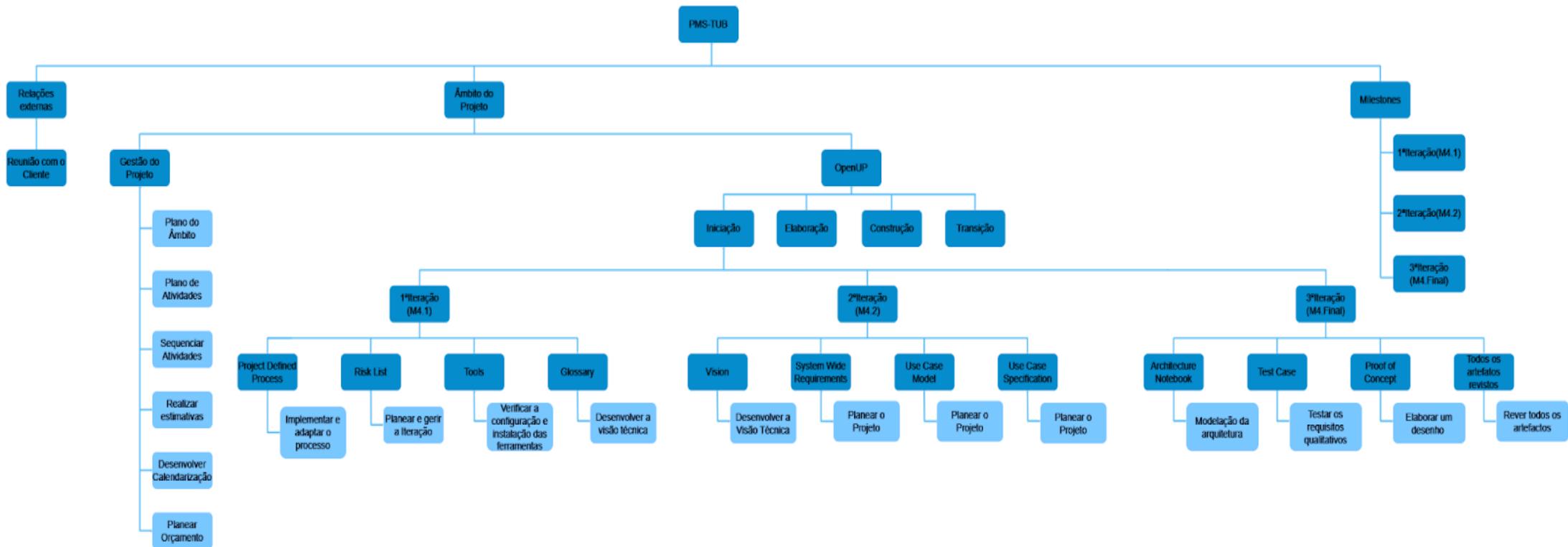
Relatório	Público-Alvo	Como é Criado / Onde é Armazenado
Relatório de Progresso	Gestora de Projeto	<ul style="list-style-type: none"> <li>Gerado automaticamente pela ferramenta de gestão de projetos;</li> <li>Armazenado no sistema de documentação central.</li> </ul>
Relatório de Qualidade	Programadores	<ul style="list-style-type: none"> <li>Criado após cada ciclo de testes;</li> <li>Armazenado na plataforma de gestão de qualidade.</li> </ul>
Relatório de Riscos	Gestora de Projeto, Stakeholders	<ul style="list-style-type: none"> <li>Elaborado pela gestora de projeto;</li> <li>Armazenado na pasta de gestão de riscos.</li> </ul>
Relatório de Desempenho do Sistema	Gestora de Projeto, Engenheiro de Projeto	<ul style="list-style-type: none"> <li>Gerado por ferramentas de monitorização de desempenho;</li> <li>Armazenado no servidor de análise técnica.</li> </ul>
Relatório de Testes	Engenheiro de Projeto, Programadores	<ul style="list-style-type: none"> <li>Gerado após a fase de testes;</li> <li>Armazenado na ferramenta de gestão de testes e na documentação do projeto.</li> </ul>
Relatório Financeiro	Gestora de Projeto	<ul style="list-style-type: none"> <li>Preparado pelo departamento financeiro;</li> <li>Armazenado numa área restrita do sistema financeiro.</li> </ul>
Relatório de Conformidade	Engenheiro de Projeto, Stakeholders	<ul style="list-style-type: none"> <li>Elaborado após auditorias de conformidade;</li> <li>Armazenado no repositório de conformidade do projeto.</li> </ul>

## 6. Papéis

A abordagem orientada por papéis fomenta uma nova forma de desenvolvimento de software, centrada maioritariamente na interação e colaboração entre as pessoas.

Papéis	Elementos do Grupo	Responsabilidades
Gestora de Projeto	Inês Melo	<ul style="list-style-type: none"> <li>• Liderança e Visão do Projeto;</li> <li>• Planeamento e Acompanhamento do Projeto;</li> <li>• Gestão de Riscos;</li> <li>• Comunicação e Coordenação;</li> <li>• Definição de Estratégia e Prioridades;</li> <li>• Inspiração e Motivação da Equipa;</li> <li>• Garantia de Prazos e Qualidade.</li> </ul>
Analistas	João Ribeiro, Inês Melo	<ul style="list-style-type: none"> <li>• Ligação entre Cliente e Equipa;</li> <li>• Compreensão de Necessidades e Requisitos;</li> <li>• Recolha de Informações (Entrevistas e Análise de Dados);</li> <li>• Garantia de Relevância do Produto para o Cliente;</li> <li>• Alinhamento do Produto com as Expectativas do Cliente.</li> </ul>
Arquiteto	Afonso Leite	<ul style="list-style-type: none"> <li>• Definição da Estrutura do Sistema;</li> <li>• Tomada de Decisões Técnicas e Estratégicas;</li> <li>• Design e Implementação do Software;</li> <li>• Equilíbrio entre as Necessidades e os Requisitos dos Stakeholders;</li> <li>• Avaliação e Gestão de Riscos Técnicos;</li> <li>• Comunicação Clara das Decisões à Equipa;</li> <li>• Garantia de Implementação das Decisões no Projeto.</li> </ul>
Programadores	João Ribeiro, Inês Melo	<ul style="list-style-type: none"> <li>• Criação de Componentes do Sistema;</li> <li>• Implementação de Funcionalidades.</li> </ul>
Tester	Pedro Lopes	<ul style="list-style-type: none"> <li>• Coordenação das Atividades de Testes;</li> <li>• Identificação, Definição e Implementação de Testes;</li> <li>• Execução dos Testes e Análise dos Resultados;</li> <li>• Configuração e Instalação de Ferramentas de Testes;</li> <li>• Suporte Técnico ao Projeto.</li> </ul>

Engenheiro de Projeto	Afonso Leite	<ul style="list-style-type: none"><li>• Planeamento, Desenvolvimento e Gestão de Projetos de Engenharia;</li><li>• Criação de Desenhos Técnicos;</li><li>• Coordenação da Equipa de Trabalho;</li><li>• Controlo de Custos e Prazos;</li><li>• Avaliação da Viabilidade Técnica e Conformidade com as Especificações;</li><li>• Garantia de Qualidade, Segurança e Sustentabilidade do Projeto;</li><li>• Comunicação com as Partes Envolvidas.</li></ul>
Especialista de Ferramentas	Pedro Lopes	<ul style="list-style-type: none"><li>• Suporte a Ferramentas Técnicas do Projeto;</li><li>• Seleção e Aquisição de Ferramentas;</li><li>• Configuração e Implementação de Ferramentas;</li><li>• Garantia de Funcionamento Adequado das Ferramentas.</li></ul>



## 7. Diretrizes e Procedimentos Específicos do Projeto

Esta secção apresenta as diretrizes e os procedimentos específicos do projeto que não fazem parte dos processos padrão da organização. Serão incluídos aqui procedimentos especiais de revisão, diretrizes de estilo e de codificação, assim como outras práticas ajustadas às necessidades específicas deste projeto. Estas orientações têm como objetivo guiar a equipa para assegurar consistência, qualidade e alinhamento com os objetivos estabelecidos, complementando as práticas organizacionais existentes.

Diretriz ou Procedimento	Responsável	Utilizado por	Local de Armazenamento
Planeamento de Iterações para o Sistema de Bilhética	Gestora de Projeto	Programadores	Documentação do Planeamento
Gestão de Riscos Técnicos e de Segurança	Gestora do Projeto	Todos os Membros	Risk List
Revisão de Artefactos com foco na Bilhética	Engenheiro de Projeto	Engenheiro de Projeto	Relatórios de Revisão
Conformidade com RGPD e Segurança de Dados	Analistas	Todos os Membros	Política de Segurança de Dados
Utilização de Ferramentas para Simulações e Testes	Tester	Tester, Programadores	Documentação dos Testes

## 2. Risk List

Após a análise inicial do Desafio 5 acerca de um Sistema de Bilhética e Ajuda à Exploração que aborda questões relacionadas com localização em tempo real, gestão de viaturas, informação ao público, monitorização e fiscalização entre outros aspectos, identificamos os principais riscos que poderão influenciar o desempenho da nossa equipa no contexto do projeto, esta análise teve como objetivo antecipar potenciais dificuldades que poderão surgir ao longo do desenvolvimento do mesmo. Os riscos identificados são cruciais para o nosso desempenho ao longo da execução do projeto, uma vez que cada um apresenta uma probabilidade de ocorrência e um impacto significativo.

Apresentamos uma tabela com os principais riscos identificados pelo nosso grupo de trabalho. Estes, foram selecionados com base na sua probabilidade de ocorrência e no impacto que poderão ter, caso se concretizem, à medida que avançamos na execução do projeto. Cada risco será acompanhado por uma breve descrição e propostas de ações atenuantes para mitigar os seus efeitos.

A escala de risco considera a probabilidade, o impacto e a magnitude para orientar as ações a serem tomadas.

A probabilidade varia de 1 (muito improvável) a 5 (muito provável), o impacto vai de 1 (muito baixo) a 5 (muito alto). A magnitude, segue uma escala de 1 a 25: de 1 a 5 é considerado baixo, exigindo apenas controlo; de 6 a 10 é moderado, sendo necessário planejar ações; de 11 a 15 é alto, recomendando-se uma ação proativa; e de 16 a 25 é muito alto, requerendo ação urgente de alta prioridade.

O risco mais crítico para o nosso grupo é a cibersegurança e ameaças externas, isto porque um ataque cibernético pode comprometer dados sensíveis dos utilizadores, prejudicar a confiança no sistema e até levar a penalidades financeiras e legais, a crescente sofisticação das ameaças cibernéticas faz com que sistemas ligados a dados e localizações em tempo real sejam alvos potenciais de hackers e mesmo com boas práticas de segurança e atualizações, a segurança cibernética nunca é garantida a 100%, sendo necessário monitorizar constantemente e investir em defesas robustas.

Este risco destaca-se pelo potencial de impacto disruptivo tanto para o funcionamento do sistema quanto para a imagem do projeto, afetando diretamente os dados dos utilizadores e a continuidade do serviço.

ID	Risco	Descrição	Tipo	Probabilidade Inicial (1-5)	Impacto Inicial (1-5)	Magnitude Inicial (= P*I)	Ações de Mitigação	Probabilidade Final	Impacto Final	Magnitude Final	Comentários
1	<b>Conformidade com Regulações de Privacidade e Segurança</b>	O sistema deve estar em conformidade com regulamentos como o RGPD, exigindo proteções para dados pessoais. Violações podem resultar em sanções legais e perda de confiança do público.	Conformidade	3	5	15	Consultar reguladores de conformidade e realizar auditorias periódicas. Implementar políticas de segurança de dados e fornecer formação contínua para a equipa.	2	4	8	Este é um risco central, dada a importância da proteção de dados. Medidas preventivas robustas podem minimizar o risco de violações e assegurar a confiança dos utilizadores.
2	<b>Qualidade dos Dados e Atualização em Tempo Real</b>	O sistema deve garantir que os dados sobre localização, horário e status dos veículos estejam atualizados em tempo real para otimizar o atendimento.	Técnico	4	4	16	Adotar tecnologias de backup e verificar regularmente a integridade dos dados. Monitorizar e ajustar algoritmos de atualização de dados em tempo real.	3	3	9	Manter a precisão dos dados exige supervisão contínua. Qualquer falha em atualizações afeta diretamente a satisfação e a segurança do utilizador.

3	<b>Cibersegurança e Ameaças Externas</b>	O sistema pode ser alvo de ataques cibernéticos que comprometem a segurança dos dados e a integridade do serviço.	Segurança	4	5	20	Implementar firewalls, criptografia de dados e sistemas de deteção de intrusões. Atualizar regularmente o sistema e realizar testes de segurança.	3	4	12	O investimento em segurança é essencial para proteger o sistema contra ameaças. Ataques cibernéticos podem prejudicar a confiança e funcionamento do serviço.
4	<b>Erros na Interface com o Utilizador</b>	Interfaces mal projetadas podem levar a frustração do utilizador, dificultando o uso do sistema de bilhética e as funcionalidades de ajuda à exploração.	Técnico	3	4	12	Conduzir testes de usabilidade e recolher feedback dos utilizadores. Investir em design intuitivo e formação de equipa para melhoria contínua.	2	3	6	Assegurar uma interface intuitiva reduz problemas de usabilidade e aumenta a satisfação do utilizador.
5	<b>Condições Meteorológicas Adversas</b>	Clima severo pode afetar a precisão da localização, conectividade e até a mobilidade dos veículos.	Externo (Ambiental)	3	4	12	Implementar redundância nos sensores e proteções físicas para dispositivos expostos. Desenvolver protocolos de emergência para cenários meteorológicos severos.	2	3	6	Clima imprevisível exige uma estratégia de redundância. Um sistema bem protegido mantém a operação mesmo em condições adversas.

6	<b>Problemas na adaptabilidade</b>	Criação de várias versões do sistema para diferentes plataformas (tablet, computador, smartphone) aumenta a complexidade de desenvolvimento e manutenção. Cada plataforma pode ter requisitos específicos de interface, desempenho e compatibilidade.	Técnico	4	3	12	<b>Adotar um Framework Multiplataforma:</b> Utilizar um framework que permita o desenvolvimento de uma única base de código com ajustes para cada plataforma, como React Native ou Flutter. <b>Design Modular:</b> Estruturar o sistema com componentes reutilizáveis, permitindo adaptações mais simples entre dispositivos. <b>Testes de Usabilidade por Plataforma:</b> Realizar testes específicos para verificar a compatibilidade e a experiência do utilizador em cada tipo de dispositivo.	3	2	6	A utilização de um framework multiplataforma simplifica o desenvolvimento e a manutenção ao longo do tempo, permitindo uma melhor alocação de recursos e uma experiência de utilização mais consistente entre dispositivos.
---	------------------------------------	---	---------	---	---	----	---	---	---	---	---

7	<b>Falta de experiência na área de negócio</b>	O grupo pode enfrentar dificuldades a interpretar corretamente as necessidades e expectativas do setor de transporte público.	Humano	4	3	12	Estudar o setor e consultar especialistas para alinhar as decisões do projeto. Participar em formações e realizar entrevistas com stakeholders do setor.	3	2	6	Desenvolver uma compreensão sólida do setor reduz a margem de erro e garante que o projeto corresponda às expectativas reais.
8	<b>Inexperiência em planear projetos</b>	A falta de experiência em gestão de projetos pode dificultar a alocação de recursos e o cumprimento dos prazos.	Gestão	5	4	20	Usar frameworks de gestão de projetos e dividir tarefas em etapas claras. Procurar orientação de um mentor ou professor para auxiliar no planeamento.	3	3	9	A gestão eficaz é fundamental para o sucesso do projeto. Repartir o trabalho e obter orientação reduz significativamente o impacto da inexperiência.

Nota: **RGPD** - Regulamento Geral sobre a Proteção de Dados (é um regulamento da União Europeia aplicado em todos os países membros).

## 3. Tools

Este artefacto consiste na especificação e descrição de todas as ferramentas que são necessárias para apoiar e suportar todo o desenvolvimento do software.

### 3.1 System STAR

O System Star é uma ferramenta usada para a avaliação dos custos de um dado projeto, baseada no modelo COCOMO. Ela permite realizar uma avaliação detalhada das características do projeto, preparando os usuários para eventuais imprevistos que possam acontecer ao longo da execução do projeto. A ferramenta oferece uma visão clara do esforço necessário para desenvolver o projeto, incluindo os custos de recursos e de tempo de cada fase do OpenUP.

### 3.2 ProjectLibre

O ProjectLibre é uma ferramenta de gestão de projetos de código aberto, ideal para planear, organizar e controlar os diferentes aspectos do projeto. Permite criar uma estrutura de tarefas detalhadamente, possibilitando também a definição de prazos e o acompanhamento das datas de início e fim, além de mostrar dependências entre as atividades. Também permite atribuir recursos, como equipas e equipamentos, auxiliando na distribuição do trabalho de forma eficiente e evitando sobrecargas. O ProjectLibre oferece algumas visualizações dos gráficos de Gantt, que facilitam o acompanhamento do progresso e permitem fazer ajustes no cronograma, além de funcionalidades para controlar os custos, tornando assim o ProjectLibre uma boa alternativa ao Microsoft Project de forma gratuita.

### 3.3 Google Docs

No contexto deste projeto, o Google Docs é uma ferramenta colaborativa fundamental para criar, editar e partilhar documentos em tempo real com a equipa de trabalho. Esta ferramenta permite também que a equipa trabalhe simultaneamente nos documentos, facilitando a documentação do projeto, planos de trabalho e cronogramas. Com a colaboração ao vivo, cada membro pode ver as alterações e adicionar comentários, melhorando a comunicação, evitando assim a duplicação de versões. Os documentos ficam armazenados na nuvem, permitindo o acesso remoto em qualquer dispositivo, mantendo um histórico de revisões de maneira a acompanhar as mudanças. Além disso, o Google Docs integra-se com outras ferramentas do Google Workspace, como o Google Sheets e o Google Drive, centralizando o trabalho e facilitando o acesso a todos os recursos do projeto num único local.

### 3.4 Google Sheets

O Google Sheets é uma ferramenta de folhas de cálculo online que permite criar, editar e gerir dados de forma colaborativa em tempo real, podendo ser acedida ao mesmo tempo pelos membros da equipa de trabalho. É ideal para organizar informações, realizar cálculos, criar gráficos e fazer análises de dados de forma eficiente. A ferramenta também oferece funcionalidades como a criação de fórmulas, filtros, tabelas dinâmicas e integração com outras ferramentas do Google Workspace, sendo a funcionalidade de tabelas a mais significativa pois consegue auxiliar-nos em alguns aspetos do projeto como por exemplo a construção de uma lista de riscos.

### 3.5 OpenUP

O OpenUP é um processo ágil e flexível que aplica abordagens iterativas e incrementais num ciclo de vida estruturado. Organiza o esforço do projeto em micro-incrementos, unidades de trabalho mais curtas com o objetivo de garantir um progresso constante e colaborativo. O projeto é dividido em algumas iterações (intervalos planeados) com o objetivo de entregar algum valor de forma previsível. O ciclo de vida do projeto é dividido em quatro fases: Iniciação, Elaboração, Construção e Transição, (Inception, Elaboration, Construction, Transition), proporcionando visibilidade e pontos de decisão ao longo do desenvolvimento do projeto, até ao lançamento final da aplicação.

### 3.6 CMMi

O CMMi baseia-se num modelo internacional de boas práticas para o desenvolvimento e manutenção de projetos, com o principal objetivo de auxiliar empresas de TI na implementação de melhorias nos processos de desenvolvimento de serviços, bem como na produção de software com maior qualidade e menos erros.

### 3.7. Visual Studio Code

O Visual Studio Code (VSCode) é um editor de código, capaz de suportar várias linguagens de programação. É altamente personalizável, possuindo extensões capazes de adicionar funcionalidade. Disponível em várias plataformas, sendo este editor de código ideal para desenvolvedores que trabalham em projetos independentemente da sua dimensão.

## 4. Glossary

Um glossário é um documento que reúne as definições dos principais termos utilizados em todas as fases do projeto, garantindo que tanto os stakeholders como os membros da equipa tenham uma compreensão clara e uniforme. Esta centralização dos termos, abreviações e definições específicas para o contexto do projeto minimiza falhas de comunicação e interpretações erradas. Além disso, proporciona uma referência útil para que os diferentes stakeholders possam aplicar os termos corretamente em diversos contextos e aspetos do projeto.

### A:

AFC (Automatic Fare Collection): Sistema automático desenhado para cobrar as tarifas dos passageiros.

API (Application Programming Interface): Conjunto de protocolos e definições que permite a comunicação e a interação entre serviços, dando assim à criação de aplicações de software.

Ambiente real: O contexto operacional onde o sistema será utilizado pelos os utilizadores finais.

Artefacto: Qualquer modelo, documento, diagrama ou outro item criado durante o ciclo de vida do desenvolvimento do projeto, utilizado para apoiar na construção bem como na existência de evidências do trabalho que está a ser realizado.

### B:

Backup: Processo de cópia e armazenamento de dados em local seguro para garantir a sua recuperação em caso de perda.

Bilhética: Conjunto de sistemas e tecnologias utilizados para a gestão, emissão, venda e validação de bilhetes em serviços como transportes públicos, eventos culturais, espetáculos ou qualquer outro tipo de entrada paga. A bilhética envolve processos que garantem a eficiência e segurança na distribuição de bilhetes, podendo incluir funcionalidades de controle de acesso e integração com diferentes formas de pagamento.

BRT(Bus Rapid Transit): Sistema de transporte público que utiliza autocarros para fornecer serviços de alta capacidade, alta frequência e paragem limitada.

### C:

Cartão da Cidade myBraga: Modelo conceitual de cartão a ser usado nos transportes de Braga, integrando as funcionalidades da bilhética digital.

CMMI (Capability Maturity Model Integration): Modelo internacional de boas práticas para desenvolvimento de software e processos, focado na melhoria contínua e na qualidade do produto.

Cloud: Rede extensa de servidores remotos em todo o mundo.

COCOMO (Constructive Cost Model): Modelo utilizado para estimar custos, o esforço e o cronograma de desenvolvimento de projetos de software com base em métricas como tamanho do código.

Construction: Fase do ciclo de vida do projeto em que o sistema é efetivamente construído e tem como principal ponto de foco a implementação, os testes realizados no software e o desenvolvimento dos requisitos.

**D:**

Diagramas/Gráficos de Gantt: Gráfico utilizado para planear e controlar o cronograma de projetos, mostrando tarefas ao longo do tempo. Representa visualmente a duração, a sequência e as dependências das atividades.

**E:**

Elaboration: Fase do ciclo de vida do projeto onde os requisitos são detalhados mais cuidadosamente. Nesta segunda das quatro fases, é feita também uma análise aos riscos do projeto.

**G:**

Google Docs: Aplicação web de processamento de texto do Google Workspace que permite criar, editar e colaborar em documentos em tempo real, com formatação, comentários e integração com outras ferramentas Google.

Google Sheets: Aplicação web de folhas de cálculo do Google Workspace que permite criar, editar e colaborar em tempo real, com fórmulas, gráficos e integração com outras ferramentas Google.

**I:**

Inception: É a fase inicial do projeto que tem como objetivo a definição dos requisitos, as metas que este pretende atingir e ainda a viabilidade do mesmo. Serve também para planear as próximas partes.

IDE: Ambiente de Desenvolvimento Integrado (IDE) é um software que oferece um conjunto de ferramentas abrangentes para facilitar o desenvolvimento de aplicações. Combina em uma interface gráfica única funcionalidades essenciais, como um editor de código-fonte que destaca a sintaxe e sugere autocompletar, ferramentas de automação para simplificar tarefas de compilação e testes, e um depurador (debugger) que auxilia na

detecção e correção de erros no código. O objetivo de um IDE é otimizar o fluxo de trabalho dos desenvolvedores, aumentando a eficiência e a produtividade.

**L:**

Layout: Disposição visual e organização de elementos em um espaço, como texto e imagens, para melhorar a funcionalidade e a apresentação.

**O:**

OpenUP (Open Unified Process): Um processo de desenvolvimento de software que segue uma abordagem ágil e iterativa, focando-se na entrega de valor de forma contínua e ajustando-se às necessidades do projeto. Promove a colaboração entre a equipa e assegura que cada fase contribui para alcançar os objetivos definidos.

**P:**

PBS (Product Breakdown Structure): Estrutura hierárquica que divide um produto em seus componentes e subcomponentes, facilitando o planeamento e a gestão do projeto.

PMBOK (Project Management Body of Knowledge): Conjunto de boas práticas e diretrizes para a gestão de projetos, abordando áreas como escopo, tempo, custo e qualidade.

ProjectLibre: Software de gestão de projetos que permite planeamento, monitorização e controlo de tarefas, cronogramas e recursos.

**R:**

RBS (Resource Breakdown Structure): Estrutura que organiza os recursos necessários para um projeto em categorias e subcategorias, de modo a facilitar a sua gestão e controlo.

Requisito: É uma condição ou capacidade que um sistema, produto ou serviço deve ter para atender às necessidades de um utilizador ou de um stakeholder. No contexto de desenvolvimento de software, os requisitos são declarações claras sobre o que o sistema deve fazer, como se deve comportar e quais as características que deve ter.

RGPD (Regulamento Geral sobre a Proteção de Dados): Legislação da União Europeia que regula o processamento de dados pessoais e visa proteger a privacidade dos indivíduos, impondo requisitos rigorosos para a recolha, armazenamento e uso de informações pessoais.

**S:**

Sistema: Conjunto de elementos interdependentes que trabalham em conjunto em prol de uma função ou objetivo específico.

Stakeholders: Pessoas, grupos ou entidades que têm interesse ou influência no projeto, podendo ser afetados pelos seus resultados ou pelo seu progresso. Incluem clientes, utilizadores finais, membros da equipa, patrocinadores e outros que contribuem para ou são impactados pelo sucesso do projeto.

SystemStar: Ferramenta de apoio à gestão e estimativa de projetos de software, com funcionalidades para calcular custos, esforços e cronogramas, utilizando modelos de estimativa como o COCOMO.

Sistema de Bilhética e Ajuda à Exploração: Questões relacionadas com localização em tempo real, gestão de viaturas, informação ao público, monitorização e fiscalização entre outros aspectos.

**T:**

Transition: É a quarta e última fase do ciclo de vida do desenvolvimento do sistema, onde o mesmo é preparado para ser entregue e consequentemente ser utilizado pelos clientes.

**U:**

Utilizador: Pessoas ou organizações que utilizam/interagem com um determinado serviço e que podem fornecer entradas bem como receber saídas, tendo, portanto, um papel ativo.

UML: Unified Modeling Language é uma linguagem gráfica para visualizar, especificar, construir e documentar os artefactos de um sistema intensivo de software.

**V:**

VSCode: Visual Studio Code é um editor de código redefinido e otimizado para criar e aprimorar aplicações modernas para web e nuvem.

**W:**

WBS (Work Breakdown Structure): Estrutura hierárquica que divide um projeto em partes menores e geríveis, facilitando a planificação e o controlo de tarefas.

## 5. Vision

### 1. Introdução

A cidade de Braga tem vindo a investir em tornar-se uma referência nacional e internacional em mobilidade sustentável e inovadora. Neste contexto, o projeto BRT (Bus Rapid Transit) representa uma oportunidade única para modernizar e digitalizar completamente o sistema de bilhética e monitorização dos transportes públicos na cidade. O objetivo é desenvolver uma aplicação móvel que ofereça uma experiência de utilizador integrada e eficiente, alinhada com a estratégia de mobilidade 4.0 da cidade.

A solução proposta abrange desde a compra e recarga de bilhetes digitais até à monitorização da ocupação dos autocarros em tempo real, oferecendo funcionalidades inovadoras para residentes, visitantes, motoristas e gestores de transporte.

### 2. Posicionamento

#### 2.1. Declaração do Problema

The problem of	Sistemas de bilhética e gestão de transporte desatualizados e ineficientes que não suportam uma experiência digital moderna.
Affects	Passageiros, turistas, motoristas, gestores da TUB e fiscalizadores.
The impact of which is	Atrasos na compra de bilhetes, dificuldade em gerir o fluxo de passageiros, baixa eficiência operacional, e insatisfação dos utilizadores.
A successful solution	Uma aplicação que permita bilhética digital, registo de ocupação em tempo real, e acesso imediato às informações de localização dos autocarros, melhorando a eficiência e a experiência do utilizador.

#### 2.2. Declaração de Posicionamento do Produto

For	Passageiros locais, turistas, motoristas e gestores da TUB.
Who	Necessitam de um sistema de bilhética moderno e de gestão de transporte eficiente e seguro.
The MobilPMS	Aplicação de Bilhética e Gestão de Transporte da TUB.
That	Oferece bilhética digital, gestão integrada de cartões como o "Cartão da Cidade myBraga", e monitorização em tempo real da ocupação dos autocarros.

<b>Unlike</b>	Soluções tradicionais que exigem tempo e não proporcionam atualizações em tempo real ou uma experiência digital integrada.
<b>Our Product</b>	Garante uma experiência de utilizador fluida, acesso fácil a informações, integração com a plataforma nacional de bilhética, e suporte para múltiplos métodos de pagamento, como MB Way, PayPal, Revolut, e pagamentos físicos.

### 3. Descrição dos Stakeholders

#### 3.1. Resumo dos Stakeholders

Nome	Descrição	Responsabilidades
<b>Passageiros</b>	Utilizadores regulares e turistas	Validar bilhetes digitais, recarregar o "Cartão da Cidade myBraga", e aceder a informações de ocupação e localização em tempo real.
<b>Motoristas</b>	Operadores dos autocarros	Utilizar a aplicação para registar entradas/saídas de passageiros, reportar incidentes, e facilitar o fluxo de passageiros.
<b>Gestores da TUB</b>	Responsáveis pela gestão da operação	Monitorizar o sistema em tempo real, gerir a frota, otimizar as rotas, e assegurar a qualidade do serviço.
<b>Fiscalizadores</b>	Equipa que verifica a conformidade	Verificar bilhetes digitais, garantir o cumprimento das normas de utilização, e monitorizar a ocupação dos autocarros.
<b>Câmara Municipal de Braga</b>	Entidade governamental responsável pela gestão pública e infraestrutura urbana.	Garantir o financiamento e suporte para a implementação do sistema. Supervisionar a integração do sistema com as políticas municipais. Promover a utilização do sistema junto da população.

<b>Consultores</b>	Docentes da UC	Fornecer orientação técnica e estratégica durante o desenvolvimento do sistema. Validar requisitos técnicos e funcionais do projeto. Auxiliar na avaliação de soluções tecnológicas e melhores práticas.
<b>Fornecedores</b>	Empresas que fornecem infraestrutura tecnológica, como hardware e software necessários para o sistema.	Garantir a entrega de dispositivos e infraestrutura para o funcionamento do sistema. Oferecer suporte técnico e manutenção dos equipamentos. Assegurar a compatibilidade entre sistemas existentes e novos.

### 3.2. Ambiente do Utilizador

O sistema será utilizado em diferentes ambientes, adaptado às necessidades específicas de vários tipos de utilizadores, e desenvolvido com considerações importantes sobre conectividade, segurança de dados e desempenho.

Os passageiros utilizarão a aplicação principalmente enquanto estão em movimento, seja em espaços urbanos ou nos próprios autocarros. Para estes utilizadores, é essencial que a aplicação ofereça acesso rápido e fácil à bilhética digital, além de atualizações em tempo real sobre a localização dos autocarros e informações sobre a ocupação. A aplicação será compatível com smartphones, suportando tanto a plataforma Android como iOS.

Os motoristas terão uma interface otimizada e dispositivos leitores de QR code dos bilhetes fixados nos autocarros, o que lhes permitirá reportar incidentes de forma simples, gerir as entradas e saídas de passageiros e aceder a informações sobre a rota de maneira intuitiva e eficaz.

Os gestores da TUB utilizarão o sistema em computadores, em ambientes de escritório, com o objectivo de monitorizar a frota, analisar dados de utilização e optimizar as operações de transporte. Estes utilizadores terão acesso a um painel de controlo centralizado que lhes permitirá visualizar, em tempo real, tanto a ocupação dos autocarros como a sua localização.

Os fiscalizadores, por sua vez, usarão dispositivos móveis, para verificar os bilhetes e garantir o cumprimento das normas de utilização. Atuarão tanto no interior dos autocarros como em paragens movimentadas, assegurando a ordem e o cumprimento das regras do sistema de transporte.

Para garantir o funcionamento eficiente e seguro do sistema, foram consideradas questões essenciais. A aplicação deverá operar mesmo em condições de conexão limitada, com um mecanismo que permita a sincronização dos dados assim que a conexão for restaurada. Além disso, a segurança dos dados pessoais e da localização dos utilizadores será uma prioridade, estando assegurada em conformidade com o Regulamento Geral sobre a

Proteção de Dados (RGPD), que exige a proteção dos dados pessoais e a transparência quanto ao seu uso.

Em termos de desempenho, o sistema foi concebido para suportar um grande número de utilizadores, especialmente em horários de pico, sem comprometer a sua performance. A robustez do sistema também foi considerada, com o objetivo de garantir uma disponibilidade de 99,9%. Para isso, serão implementados planos de contingência adequados para lidar com eventuais falhas, assegurando que o serviço seja estável e fiável em todas as situações.

## 4. Visão Geral do Produto

### 4.1. Necessidades e características

Onde 1 é a prioridade mais alta e 5 a mais baixa.

Necessidade	Prioridade (1 a 5)	Funcionalidades	Lançamento Planeado
Gestão de Crédito de Cartões	1	Opção para recarregar o “Cartão da Cidade myBraga” através da aplicação com diversos métodos de pagamento (MB Way, PayPal, etc.).	Primeira Versão
Compra de Bilhetes	1	Compra de bilhetes únicos, tanto na aplicação, como nos autocarros, com suporte para impressão de bilhetes descartáveis.	Primeira Versão
Monitorização em Tempo Real	1	Informação sobre a ocupação dos autocarros e localização em tempo real, com contínuas atualizações.	Primeira Versão
Localização via SMS	3	Opção para os utilizadores que não possuem smartphones, enviando informações via mensagem automática.	Segunda Versão
Notificações e Alertas	3	Enviar notificações automáticas para alterações de horários ou cancelamentos.	Segunda Versão
Ferramentas para Validação	2	Hardware que permita leitura de códigos QR, presentes nos bilhetes, de forma a efetuar a validação	Segunda Versão



## 5. Outros Requisitos do Produto

Onde 1 é a prioridade mais alta e 5 a mais baixa.

Requisito	Prioridade (1 a 5)	Lançamento Planeado
Padrões e Conformidades	1	Primeira Versão
Segurança de Dados	1	Primeira Versão
Desempenho	2	Segunda Versão
Robustez e Tolerância a Falhas	1	Primeira Versão
Requisitos de Documentação	3	Primeira Versão, com atualizações contínuas

## 6. System Wide Requirements

### 1. Introdução

**System Wide Requirements** (Requisitos do Sistema) são requisitos gerais que asseguram o bom funcionamento de um sistema em ambientes amplos e complexos. Incluem compatibilidade com outros sistemas, escalabilidade para suportar crescimento, segurança de dados, desempenho eficiente, fiabilidade para operação contínua e acessibilidade para todos os utilizadores. Estes elementos garantem que o sistema atende às necessidades técnicas e de uso com estabilidade e segurança.

### 2. Requisitos Funcionais de todo o Sistema

Os System-Wide Functional Requirements definem as funcionalidades globais que o sistema deve apresentar, assegurando que todos os aspectos críticos do funcionamento são atendidos.

ID Requisito	Requisito	Necessidade	Descrição
1	Autenticação de Utilizadores	Necessidade de garantir o acesso seguro e controlado ao sistema	O sistema deve permitir que os utilizadores façam login com dados seguros e que utilizem métodos de autenticação adequados, como nome de utilizador e palavra-passe ou autenticação multifator.
2	Auditoria de registo	Necessidade de manter um registo detalhado das operações e dos acessos realizados no sistema	Todas as operações de leitura, escrita e modificação devem ser registadas com informações como data, hora, utilizador e tipo de operação realizada.
3	Emissão de Relatórios	Necessidade de oferecer funcionalidades para compra, renovação e cancelamento de bilhetes	O sistema deve fornecer funcionalidades para gerar relatórios configuráveis pelos utilizadores, permitindo a exportação em diferentes formatos
4	Gestão de bilhetes	Necessidade de oferecer funcionalidades para compra, renovação e cancelamento de bilhetes	O sistema deve permitir aos utilizadores, por exemplo, a compra de bilhetes digitais, renovação de passes mensais e o cancelamento de bilhetes, com um histórico de operações associado.
5	Monitorização em Tempo Real	Necessidade de monitorizar a localização e estado dos veículos em operação	O sistema deve ser capaz de mostrar a localização dos veículos em tempo real, permitindo aos gestores e aos utilizadores visualizar o estado de operação, horários e possíveis atrasos.

## 3. Requisitos de Qualidade do Sistema

### 3.1. Usabilidade

Os requisitos de usabilidade avaliam a interface com o utilizador e incluem cinco fatores com várias subcategorias, tais como: prevenção de erros, estética e design, ajuda, documentação, consistência e padrões.

**Compreensão e Facilitação no Uso** - É baseado na rapidez com que um utilizador sem familiaridade com a interface consegue realizar tarefas básicas. Considera se a interface está estruturada de maneira a ser fácil de utilizar para permitir um novo utilizador trabalhar eficientemente.

**Compreensibilidade na Interpretação** - Todas as mensagens e alertas do sistema devem ser claras e de fácil compreensão para o utilizador.

**Facilidade na Aprendizagem** - Após aprender a realizar uma tarefa, o utilizador deve ser capaz de se lembrar facilmente de como executá-la novamente.

**Satisfação Subjetiva** - Uma quantidade de utilizadores deve gostar e sentir-se confortável ao utilizar o sistema.

**Fluidez nas Operações** - O sistema deverá conseguir atender todas as situações rapidamente e com eficiência.

### 3.2. Fiabilidade

A fiabilidade é definida como a probabilidade de um sistema funcionar sem falhas em um ambiente específico em determinadas condições, ao longo de um período de tempo estabelecido.

**Precisão** - É fundamental que todos os cálculos realizados pelo algoritmo sejam precisos e exatos, garantindo que os resultados fornecidos pelo sistema correspondam à realidade.

**Disponibilidade** - Refere-se à necessidade de o sistema funcionar durante o máximo de tempo possível. Para garantir isso, é essencial contar com profissionais responsáveis pela manutenção programada dos servidores e com uma fonte de energia que assegure a continuidade das operações do sistema.

**Recuperabilidade** - Quando o sistema falha, é obrigatório repô-lo o mais rapidamente possível. Para isso, são necessários backups e profissionais com as competências e os recursos adequados para garantir a correção do erro, de maneira a evitar que volte a ocorrer.

**Frequência e Gravidade das Falhas** - O sistema apresenta, por vezes, erros, sendo alguns mais graves do que outros, o que torna necessária existir uma categorização dos erros de acordo com a sua severidade. Por defeito, estes são classificados como "*minor*", "*significant*" e "*critical*" (menor / significativo / crítico), em ordem crescente de gravidade.

### 3.3. Desempenho

Os requisitos de desempenho abordam tópicos como a taxa de transferência de informações através do sistema, capacidade de resposta e utilização de recursos.

**Tempo de Resposta** - Avalia o tempo de resposta entre a interação do utilizador e a interface. A resposta deverá ser rápida e eficaz garantindo um bom desempenho do sistema.

**Taxa de Transferência** - Garante que o Sistema consegue suportar um determinado fluxo de informações, sem que este entre em sobrecarga e corrompa informações ou que influencie o tempo de resposta.

**Capacidade** - O Sistema deve ter uma capacidade capaz de incluir todos os utilizadores do TUB (Transportes Urbanos de Braga) e permitir introduzir novos utilizadores no sistema, dado que existe a possibilidade de crescimento no número de passageiros.

### 3.4. Suporte

**Adaptação** - O sistema deverá conseguir ajustar-se ao tipo de utilizador que utiliza o sistema.

**Compatibilidade** - A aplicação deverá ser compatível com os diferentes dispositivos dos utilizadores (Android e IOS).

**Instalação** - Esta não deve ter qualquer restrição, qualquer pessoa que tenha ligação com os Transportes Urbanos de Braga poderá facilmente instalar e usufruir da mesma.

**Manutenção** - O sistema deverá ser capaz de passar por diversas adaptações, caso sejam necessárias.

**Escalabilidade** - O sistema deverá conseguir lidar com um grande fluxo de dados que serão introduzidos pelos diversos utilizadores sem que o seu desempenho sofra alterações e sem que o seu tempo de carregamento altere consequentemente.

### 3.5. Segurança

O sistema dos TUB deve permitir que os utilizadores possam criar uma conta de forma simples e segura. Inicialmente, o utilizador terá a opção de criar a sua conta, durante o processo, terá de associar um método de pagamento como MB WAY, Paypal, entre outros métodos de pagamento.

Após a associação dos métodos de pagamento, o sistema irá solicitar informações adicionais, como um nome, email, uma password e um número de telefone. Para garantir a legitimidade da informação fornecida, será enviado um código de verificação para o número de telefone registado. O utilizador deverá introduzir esse código no sistema, confirmando assim que o número é válido e pertence ao próprio.

Com esta verificação concluída, a conta será ativada e o utilizador poderá utilizar os serviços dos TUB de forma segura e protegida, com acesso restrito aos dados pessoais e financeiros associados.

A câmara nos autocarros deve detectar apenas os dados essenciais, como nome e número de telefone, para conseguir ser efetuado o pagamento, sem recolher informações adicionais. Esses dados devem ser encriptados durante a transmissão e armazenamento, e podem ser tokenizados para garantir que, mesmo se acessados indevidamente, não sejam legíveis. A base de dados deve ser protegida com controlo de acesso rigoroso, e os dados pessoais devem ser eliminados ou anonimizados quando não forem mais necessários.

## 4. Interfaces do Sistema

### 4.1 Interfaces do Utilizador

#### 4.1.1 Aparência e iteratividade

A interface do nosso sistema será desenvolvida para proporcionar ao utilizador uma máxima facilidade no acesso a todas as informações e funcionalidades necessárias, especialmente no que se refere à compra e gestão de bilhetes, bem como à monitorização de veículos. O layout terá um design moderno e acima de tudo intuitivo, de modo a que a utilização seja prática e eficiente tanto por parte dos passageiros como dos gestores.

Para garantir a acessibilidade e a facilidade de uso, os elementos de navegação e de interação serão amplos, bem visíveis e com resposta o mais rápida possível. Como sabemos que os utilizadores do BRT podem ser de qualquer faixa etária, isto facilitará o uso, por exemplo, por parte de utilizadores com menor condição visual ou física. Serão ainda usadas cores intuitivas que orientam as ações dos utilizadores e melhoram a experiência geral de navegação, focando-se em tornar cada interação simples e imediata.

A aplicação será altamente adaptável a diferentes dispositivos, garantindo que a experiência seja consistente e otimizada independentemente do meio utilizado. Este foco na simplicidade, que hoje em dia faz parte do contexto hodierno, assegurará que todos os utilizadores possam navegar pelo sistema de forma eficiente, sem barreiras ou dificuldades desnecessárias.

#### 4.1.2 Requisitos de Layout e Navegação

A aplicação será desenvolvida para proporcionar aos passageiros uma experiência simples, eficiente e intuitiva no acesso às funcionalidades oferecidas pelo sistema. O layout e a navegação serão desenhados para facilitar a compra e gestão de bilhetes, consulta de horários e o acesso a informações em tempo real.

Os passageiros terão uma área de autenticação para acederem à sua conta e usufruírem das funcionalidades personalizadas. O sistema será projetado para funcionar de maneira otimizada em diferentes dispositivos, como telemóveis e tablets, garantindo que os principais recursos estejam sempre acessíveis. Tendo em conta que a navegação nos meios referidos é diferente, o sistema terá layouts adaptados para cada um desses formatos.

A interface contará com menus e botões de fácil identificação, permitindo que os utilizadores encontrem rapidamente as funções necessárias, como comprar bilhetes ou consultar informações de viagem. A navegação será fluida, incluindo atalhos para operações frequentes.

As ações realizadas pelos utilizadores, como a compra de bilhetes ou a consulta de rotas, serão acompanhadas de feedback visual (por exemplo, notificações de confirmação ou de avisos), garantindo que os passageiros compreendam o estado das suas operações de forma imediata.

#### 4.1.3 Consistência

A consistência na interface será assegurada através da aplicação uniforme de elementos visuais e de padrões de interação. Todos os botões, menus e ícones seguirão um design similar e com um comportamento em diferentes partes da aplicação, garantindo que os passageiros saibam o que esperar durante a navegação. Os elementos de navegação estarão sempre posicionados de forma previsível, e as ações realizadas pelo utilizador serão feitas de uma forma fluida.

#### 4.1.4 Requisitos de Personalização do Utilizador

O sistema permitirá que os passageiros personalizem a sua experiência de utilização para melhor atender às suas necessidades e preferências. Os requisitos de personalização incluem:

- Definição de Preferências de Viagem: Os utilizadores poderão definir preferências pessoais, como rotas favoritas, horários preferidos e notificações sobre alterações nas suas rotas habituais, por exemplo.
- Configuração de Notificações: O sistema permitirá que os utilizadores escolham como desejam ser notificados sobre atualizações de horários, alertas de serviço ou promoções, seja por notificações push, e-mail ou SMS.

- Histórico de Utilização: Os passageiros terão acesso ao seu histórico de bilhetes e viagens, facilitando o acompanhamento das suas atividades e a rápida repetição de compras ou consultas frequentes.
- Personalização da Interface: Quando aplicável, os utilizadores poderão ajustar algumas opções da interface, como o tamanho da fonte ou o tema por exemplo, para melhor adaptação às suas necessidades, garantindo acessibilidade e conforto na utilização.

## 4.2 Interfaces para Sistemas Externos ou Dispositivos

### 4.2.1 Interface de software

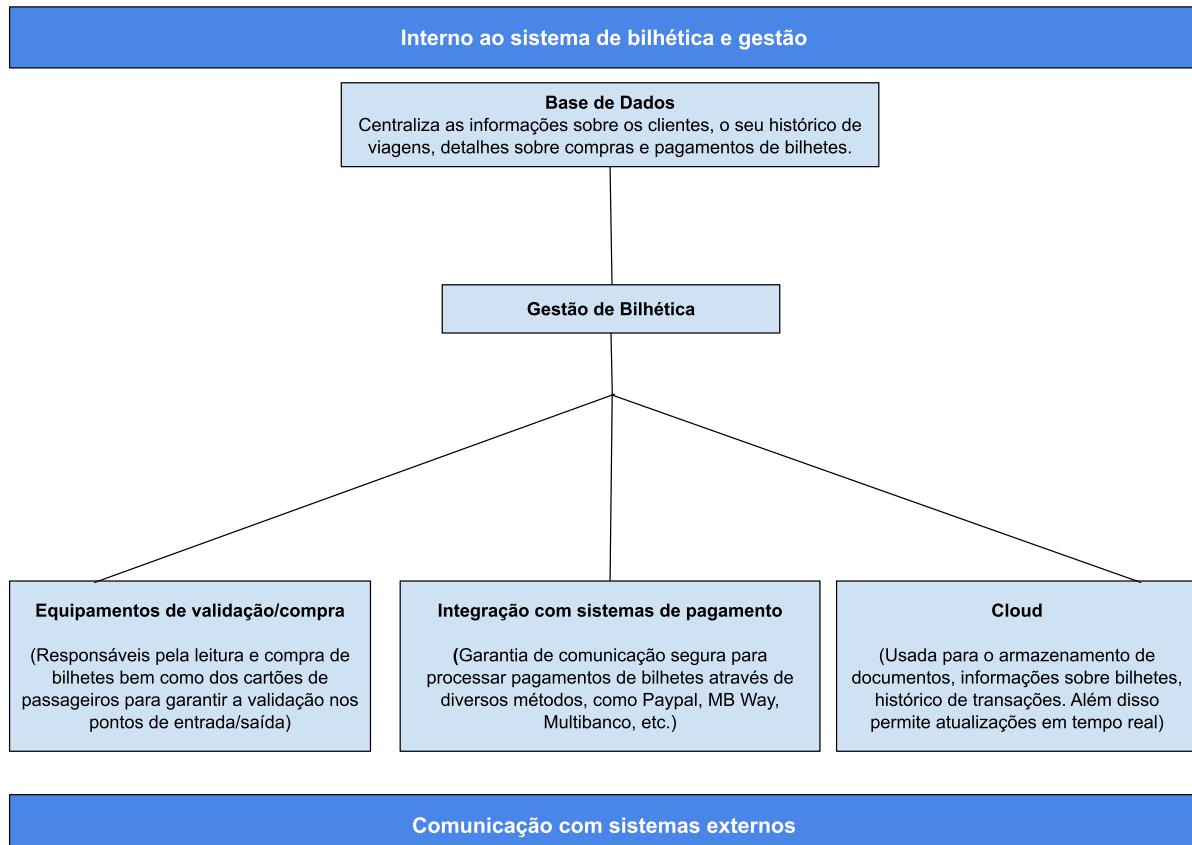
A aplicação que o estamos a desenvolver terá uma interação ampla com a base de dados central, onde estarão armazenadas as informações dos passageiros como o histórico de bilhetes, entre outros dados. Além disso, será feita a comunicação com outros sistemas, como plataformas de pagamento e serviços de monitorização.

### 4.2.2 Interfaces de Hardware

Para que a aplicação possa ser utilizada de forma integral, será necessária uma comunicação com os dispositivos de hardware, como por exemplo, as máquinas de validação ou de compra automática de bilhetes, usando o sistema AFC (Automatic Fare Collection), e assim, poder usar estas operações, entre outras, com a aplicação neste caso. Além disso, a aplicação irá integrar-se com hardware de monitorização de veículos, como sistemas de GPS instalados nos autocarros, que fornecerão informações de localização em tempo real.

### 4.2.3 Interfaces de comunicação

Para que o sistema funcione de forma integral, será necessária uma comunicação robusta com diferentes interfaces. O sistema deve garantir que todas as informações pertinentes aos passageiros sejam transmitidas ou acessadas de forma rápida, assegurando o funcionamento de serviços como a compra e validação de bilhetes, comunicação de horários, e atualizações sobre os transportes.



Cada ligação representada na imagem acima contém uma API associada permitindo, assim, a troca de informações.

#### **API entre a Gestão de Bilhética e a Base de Dados:**

Descrição: Esta API é responsável por gerir a comunicação entre o módulo de gestão de bilhética e a base de dados central. Permite o envio e a receção de dados relacionados com os clientes, bilhetes e pagamentos.

#### Principais Funcionalidades:

- Consultar histórico de viagens de um cliente.
- Verificar a validade de bilhetes no momento da utilização.
- Atualizar a base de dados com compras de novos bilhetes ou alterações no saldo do cliente.
- Sincronizar dados sobre pagamentos realizados e detalhes de transações..

### **API entre a Gestão de Bilhética e os Equipamentos de Validação/Compra:**

Descrição: Esta API permite a comunicação entre a gestão de bilhética e os equipamentos de validação/compra (como validadores e máquinas de venda). Garante o funcionamento em tempo real.

#### Principais Funcionalidades:

- Verificar, no momento da validação, se o bilhete é válido e autorizado.
- Registar nos equipamentos o histórico de utilização do bilhete.
- Enviar atualizações sobre as transações realizadas nos equipamentos para o sistema central.
- Sincronizar dados sobre cartões de clientes (recarregar saldo ou atualizar informações de perfil).

### **API entre a Gestão de Bilhética e a Integração com Sistemas de Pagamento**

Descrição: Esta API facilita a comunicação com os gateways de pagamento, permitindo que os utilizadores realizem pagamentos por diferentes métodos.

#### Principais Funcionalidades:

- Processar pagamentos de bilhetes através de plataformas como PayPal, MB Way ou Multibanco.
- Validar e autenticar os detalhes do pagamento enviados pelo cliente.
- Atualizar o sistema de bilhética com o estado do pagamento (pago, pendente, falhado).
- Criar recibos digitais após a conclusão de pagamentos.

### **API entre a Gestão de Bilhética e a Cloud**

Descrição: Esta API assegura a troca de informações entre o sistema de bilhética e o serviço cloud, garantindo a sincronização e o armazenamento seguro dos dados.

#### Principais Funcionalidades:

- Enviar e armazenar documentos, como históricos de viagens, transações e bilhetes emitidos.
- Recuperar dados armazenados na cloud para análise ou consulta.
- Permitir atualizações em tempo real no sistema de bilhética, como notificações sobre novas compras ou transações.
- Garantir redundância e backup dos dados sensíveis.

## 5. Regras de Negócio

### 5.1. Privacidade

É essencial assegurar que as informações sobre os passageiros, armazenadas na base de dados, permaneçam inacessíveis a qualquer pessoa externa, garantindo também que esses dados não sejam compartilhados com utilizadores não autorizados no sistema dos TUB.

### 5.2. Restrições

A aplicação só poderá ser acedida por passageiros ou pela equipa de manutenção da aplicação.

### 5.3. Disponibilidade

A aplicação estará disponível 24 horas, 365 dias por ano, permitindo acesso às informações a qualquer momento durante o horário de trabalho. Dessa forma, os utilizadores, incluindo aqueles que viajam para o estrangeiro, não terão problemas de acesso devido a diferenças de fuso horário.

## 6. Restrições do sistema

O desenvolvimento e operação do sistema de bilhética estão sujeitos a várias restrições que deverão ser cuidadosamente consideradas para garantir a segurança e a usabilidade da aplicação.

### Requisitos de Hardware

A aplicação deverá ter um desempenho estável independentemente do nível de processamento do equipamento que está a utilizá-la. Antes disso, o sistema deverá também ser compatível com qualquer tipo de dispositivo (tablet, telemóvel, computador, etc.).

### Ligação à internet

A operação do sistema de bilhética depende de uma conexão à internet estável para comunicação com o servidor central e armazenamento de informações na cloud. Em localidades onde a conexão seja limitada ou inexistente, o sistema deverá ser capaz de operar com funcionalidades básicas em modo off-line.

### Armazenamento

A aplicação deverá ser capaz de armazenar e processar grandes volumes de dados, como é o caso, por exemplo, do histórico de transações e os perfis de utilizador.

### Segurança

O sistema deve cumprir com as normas e regulamentos aplicáveis em relação à proteção de dados, como o Regulamento Geral sobre a Proteção de Dados (RGPD). Deve também ser implementada uma infraestrutura de autenticação para proteger os dados dos utilizadores e assim evitar acessos indesejáveis por parte de terceiros.

## Compatibilidade com Sistemas Externos

A aplicação precisará interagir com sistemas externos, como é o caso com os sistemas de pagamento. Qualquer mudança nos protocolos de comunicação ou APIs poderá impactar a operação e vão ser exigidos ajustes.

# 7. Conformidade do sistema

## 7.1 Requisitos de Licenciamento

Os dados pessoais dos utilizadores armazenados pela aplicação dos TUB serão utilizados exclusivamente para as funcionalidades relacionadas com o serviço de transporte, garantindo a proteção da privacidade de cada utilizador. O acesso às funcionalidades da aplicação será controlado e definido pelo responsável da administração dos TUB, ou por um membro autorizado da equipa de gestão, assegurando que apenas pessoas autorizadas consigam gerir e aceder a informações sensíveis.

## 7.2. Legalidades, Direitos de Autor e outros Avisos

Todos os direitos de propriedade intelectual e patentes deste projeto estão protegidos e reservados ao grupo 2 da UC de Processos e Metodologias de Software, alusivo ao curso de Engenharia e Gestão de Sistemas de Informação e pertencem aos elementos do Grupo: Afonso Leite, Inês Melo, João Ribeiro, Pedro Lopes.

A aplicação obedecerá ao Regulamento Geral sobre a Proteção de Dados garantindo que a informação dos utilizadores está protegida. Este regulamento apresenta um conjunto único de regras relativa à proteção de dados, sobre o seu tratamento e livre circulação

## 7.3. Leis Aplicáveis

Regulamento Geral sobre a Proteção de Dados (RGPD). A legalidade do sistema está garantida na Constituição da República Portuguesa, artigo 35º, parágrafo 4: "É proibido o acesso a dados pessoais de terceiros, salvo em casos excepcionais previstos na lei".

A frase mencionada "É proibido o acesso a dados pessoais de terceiros, salvo em casos excepcionais previstos na lei" está relacionada com a proteção de dados pessoais e com o direito à privacidade, que está consagrado na Constituição da República Portuguesa, especificamente no artigo 35º, como já foi referido anteriormente. Este artigo garante a proteção da privacidade e define que o acesso a dados pessoais de terceiros é restrito, salvo em situações excepcionais previstas por lei.

# 8. Documentação do Sistema

A documentação deverá incluir, em primeiro lugar, um manual detalhado sobre como utilizar a aplicação. A responsabilidade de adquirir a aplicação será dos Transportes Urbanos de Braga (TUB), que poderá instalá-la nos dispositivos que considerar adequados

e fornecê-la aos utilizadores, como passageiros, condutores e outros colaboradores que necessitem da mesma.

Além disso, será necessário fornecer descrições claras sobre como utilizar cada tipo de interface, com base nos diferentes níveis de utilizadores, como o pessoal administrativo, motoristas ou passageiros, garantindo que cada utilizador tenha acesso às funcionalidades pertinentes ao seu perfil e função.

**Passageiro** - O passageiro utiliza a aplicação para comprar bilhetes, consultar horários e rotas dos autocarros, e visualizar todo o seu histórico de viagens. A aplicação estará disponível apenas para utilizadores registados, com um número de telefone verificado, permitindo a compra de bilhetes de forma simples e prática. Com um login exclusivo, o passageiro tem acesso a informações em tempo real, incluindo notificações por mensagem para o número de telemóvel sobre as alterações de horário ou outros alertas, sendo responsável por garantir a segurança e pontualidade das viagens. Relativamente à interface, ela será intuitiva com as funcionalidades principais claramente identificadas, facilitando todo o processo.

**Administrador** - O administrador terá acesso total à aplicação para gerir utilizadores, analisar os dados dos passageiros e dos motoristas, realizar ajustes nos horários e nas rotas, além de poder monitorizar os pagamentos efetuados. Com um nível elevado de acesso, o administrador poderá fazer relatórios detalhados sobre as operações e garantir o bom funcionamento do sistema.

**Administrador Técnico (Gestor do Sistema)** - O gestor de sistema é responsável pela manutenção técnica da aplicação, incluindo a gestão de segurança, backups de dados e monitorização do desempenho. Com acesso total, ele assegura que a aplicação funcione sem problemas técnicos, implementando soluções e ajustando configurações se necessário. A interface é focada em aspectos técnicos e de segurança, garantindo a integridade e a proteção dos dados.

Em suma, identificamos a necessidade da seguinte documentação:

1. Manual do Utilizador:

- Passageiro:
  - Como criar uma conta, fazer login, recuperar senha.
  - Como comprar bilhetes e consultar históricos.
  - Como visualizar rotas e receber notificações em tempo real.
  - FAQ com problemas comuns (ex.: falha no pagamento).
- Motorista:
  - Como utilizar o scanner de bilhetes.
  - Como relatar incidentes.
  - Monitorização de entrada/saída de passageiros.
- Administrador:
  - Gerir utilizadores.
  - Ajustar horários e rotas.
  - Gerar e interpretar relatórios de utilização.

- Administrador Técnico:
  - Configurações de segurança e gestão de dados.
  - Procedimentos para backups e recuperação de dados.
  - Monitorização de desempenho e solução de problemas técnicos.

## 2. Manual Técnico:

- Documentação técnica para a equipa de desenvolvimento e manutenção.
- Especificações do Sistema:
  - Requisitos de hardware e software para instalação.
  - Arquitetura do sistema.
- Integrações:
  - APIs utilizadas para pagamentos, notificações e monitorização.
- Gestão de Erros e Logs:
  - Como interpretar logs do sistema e corrigir erros comuns.
- Plano de Manutenção:
  - Checklists para manutenção periódica (atualizações, backups, monitorização de desempenho).

## 3. Documentação de Segurança:

- Política de privacidade e proteção de dados dos utilizadores.
- Detalhamento de como as informações pessoais (ex.: dados do passageiro) são protegidas.
- Ações em caso de incidentes de segurança (ex.: fuga de dados).

## 4. Relatórios e Indicadores:

- Como gerar relatórios de utilização (número de passageiros, vendas de bilhetes, ocupação dos autocarros).
- Exemplos de KPIs que podem ser monitorizados.

## 5. Política de Atualizações:

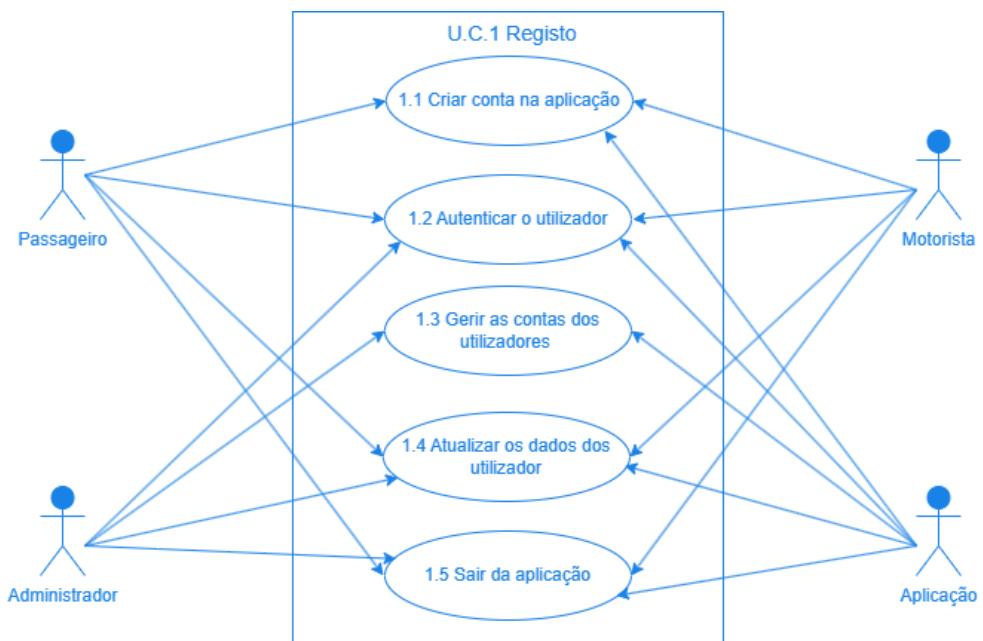
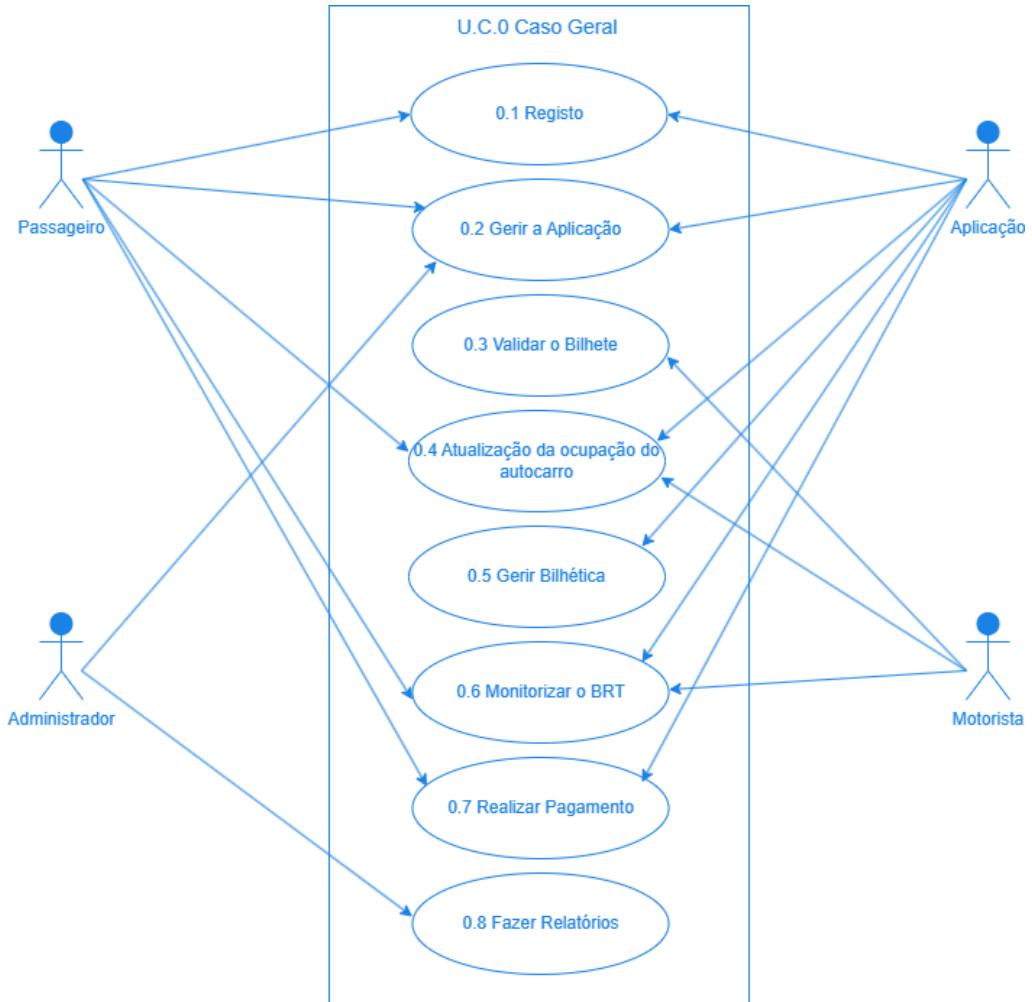
- Planeamento para introdução de novas funcionalidades.
- Garantia de suporte para versões mais antigas durante um período específico.

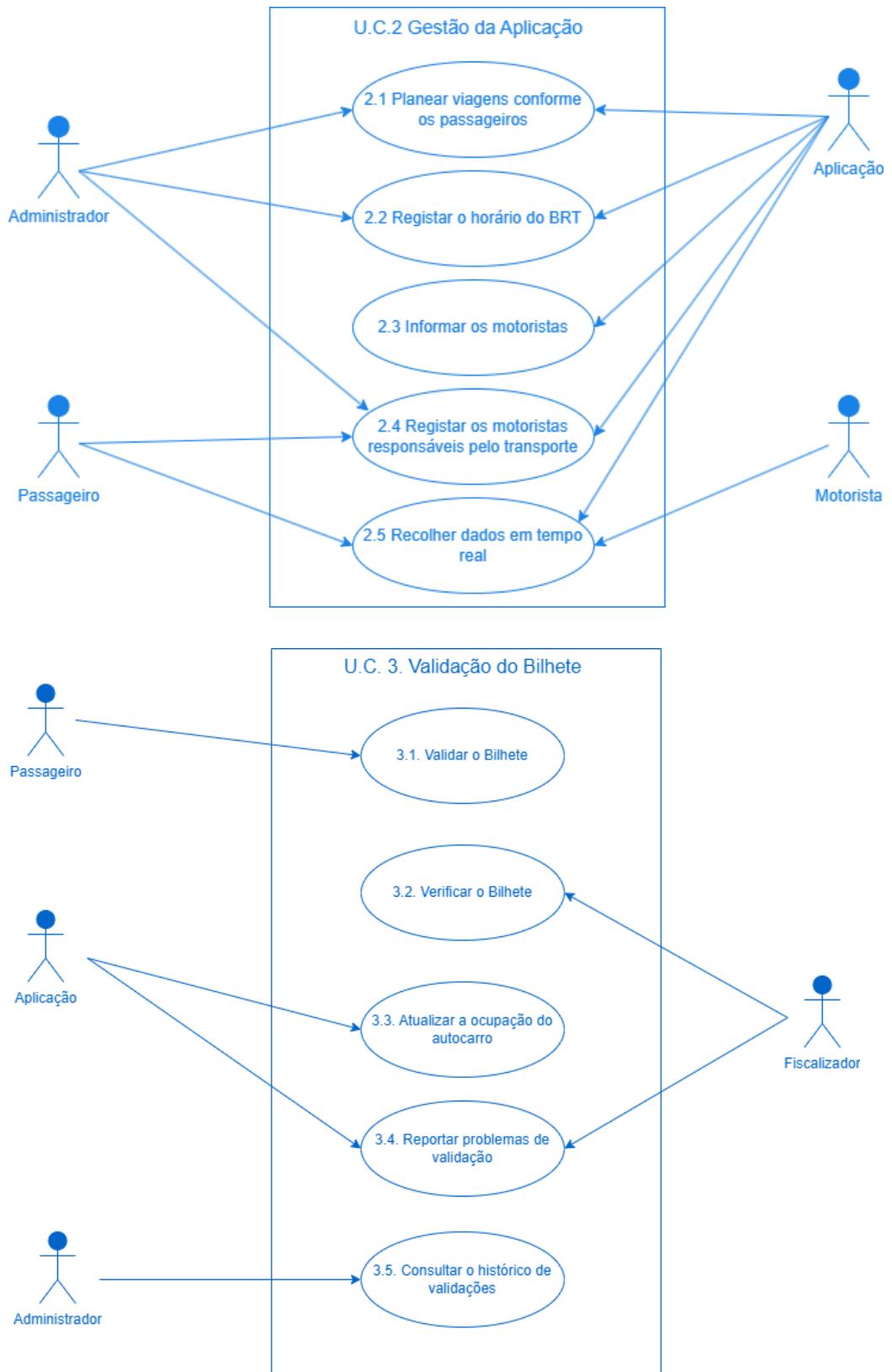
## 6. Plano de Contingência:

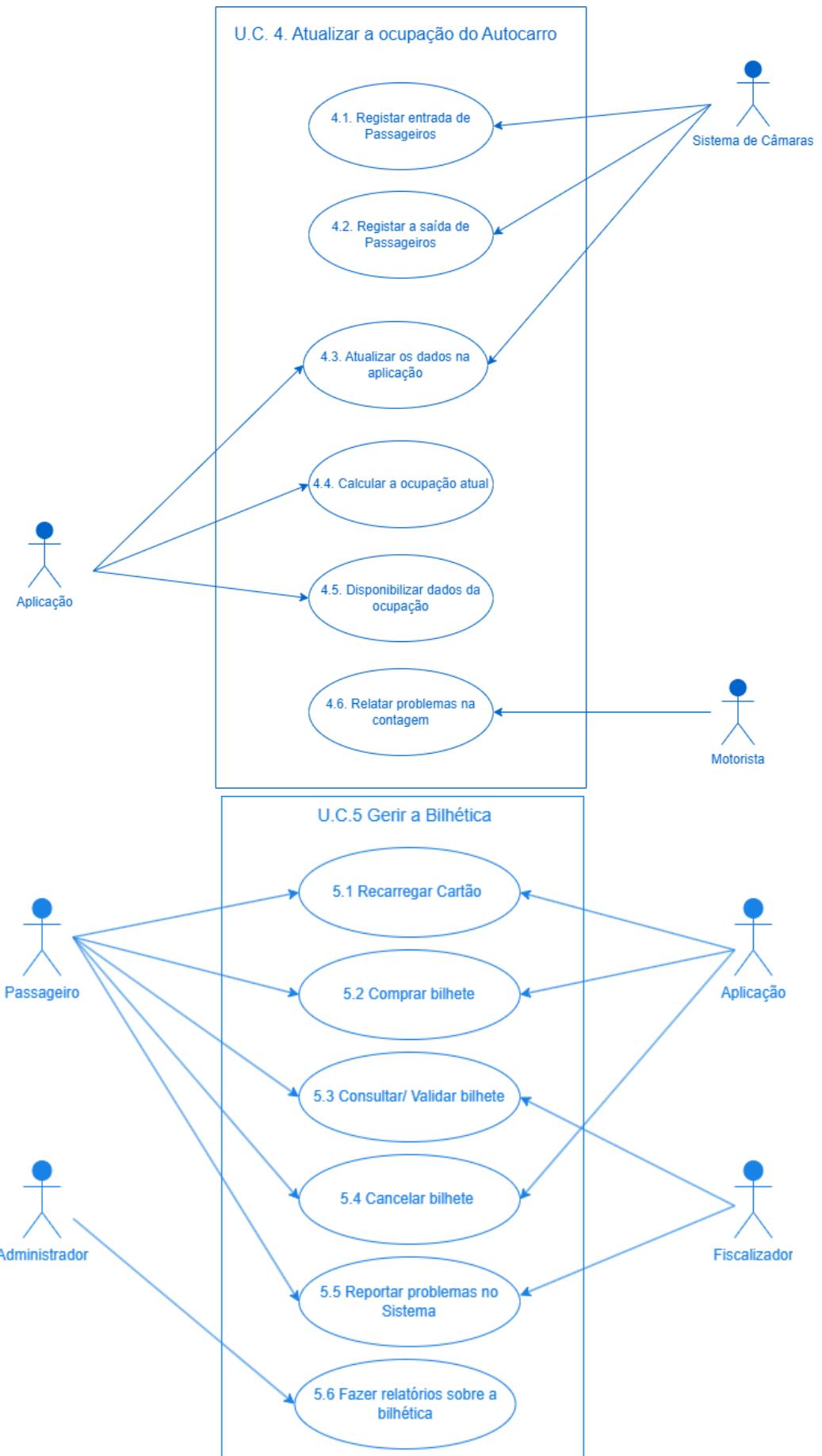
- Documentação sobre como agir em caso de falhas no sistema (ex.: falha de pagamento, indisponibilidade do servidor).
- Procedimentos para comunicação com utilizadores em situações críticas.

## 6. Use Case Model

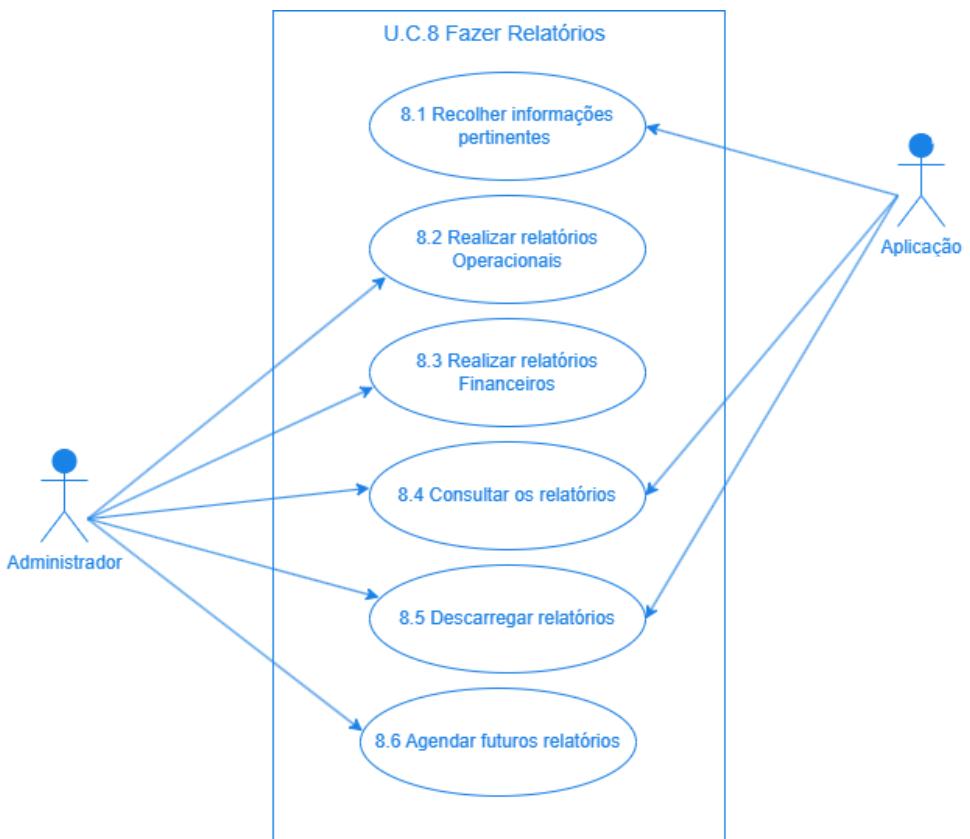
O objetivo do Use Case Model é capturar o comportamento do sistema para criar um resultado observável e valioso para os que interagem com ele.











## 7. Use Case Specification

O Use Case Specification descreve de maneira detalhada como o sistema deve responder às interações dos atores, garantindo que as funcionalidades atendam às necessidades dos utilizadores e stakeholders.

### 0.1 Registo

#### 1. Descrição

Permite ao passageiro criar uma conta na aplicação para aceder às funcionalidades personalizadas, como compra de bilhetes, monitorização de autocarros e consulta de horários.

#### 2. Atores

Passageiro	Solicita a criação de uma nova conta
Aplicação	Processa o registo.

#### 3. Pré-Condições

O utilizador deve fornecer informações válidas, como e-mail e número de telemóvel.  
O sistema deve estar operacional.

#### 4. Fluxo Básico de Eventos

O use case começa quando o passageiro seleciona "Registo".  
O passageiro insere os dados (email, nome, senha).  
A aplicação valida os dados.  
A conta é criada e o passageiro é notificado.

#### 5. Fluxos Alternativos

**Falha na validação do e-mail ou número de telemóvel**  
A aplicação informa o erro e solicita a correção dos dados.

#### 6. Fluxos Secundários

##### Validar Dados de Registo

Validar e-mail e número de telemóvel.  
Validar a palavra-passe de acordo com as políticas de segurança.

## 7. Cenários Principais

### Criar conta com dados válidos

Passageiro insere dados válidos.  
A conta é criada.

## 8. Pós-Condições

A conta é criada com sucesso e o passageiro pode acessar a aplicação.

## 9. Requisitos Especiais

Os dados devem ser criptografados.

## 0.2 Gerir a Aplicação

### 1. Descrição

Permite ao administrador ajustar configurações da aplicação e monitorizar o desempenho geral.

### 2. Atores

Administrador	Realiza configurações e supervisão.
Aplicação	Executa as alterações solicitadas.

### 3. Pré-Condições

O administrador deve estar autenticado.

### 4. Fluxo Básico de Eventos

O administrador acede ao menu "Configurações".  
Realiza ajustes, como alterar rotas ou atualizar horários.  
A aplicação guarda e aplica as alterações.

### 5. Fluxos Alternativos

#### Erro de configuração

O sistema informa o erro e solicita a correção.

### 6. Fluxos Secundários

### **Validar alterações**

Confirmar permissões do administrador.  
Aplicar configurações.

## 7. Cenários Principais

### **Alterar horários de rotas**

O administrador seleciona a rota e insere o novo horário.  
O sistema guarda as alterações.

## 8. Pós-Condições

As alterações são aplicadas com sucesso.

## 9. Requisitos Especiais

As alterações devem ser auditadas.

## **0.3 Validar o Bilhete**

### 1. Descrição

Permite a validação do bilhete de modo a garantir a sua autenticidade.

### 2. Atores

Passageiro	Apresenta o bilhete.
Aplicação	Valida o QR Code e retorna o resultado.

### 3. Pré-Condições

O bilhete deve ser válido.

### 4. Fluxo Básico de Eventos

O use case começa quando o passageiro apresenta o bilhete.  
A aplicação verifica o QR Code.  
O sistema confirma a validade.

### 5. Fluxos Alternativos

### Bilhete inválido

O sistema informa a rejeição do bilhete.

## 6. Fluxos Secundários

### Verificar validade do bilhete

Decodificar QR Code.

Consultar validade.

## 7. Cenários Principais

### Bilhete válido

O passageiro apresenta bilhete válido.

O sistema autoriza o embarque.

### Bilhete inválido

O passageiro apresenta um bilhete inválido

O sistema não autoriza o embarque

## 8. Pós-Condições

O bilhete fica marcado como usado.

## 9. Requisitos Especiais

A validação deve ser rápida (em tempo real).

## 0.4 Atualização da Ocupação do Autocarro

### 1. Descrição

Permite ao motorista atualizar o número de passageiros no autocarro.

### 2. Atores

Motorista	Regista os dados.
Aplicação	Atualiza e exibe as informações em tempo real.

### 3. Pré-Condições

O motorista deve estar autenticado.  
O sistema deve estar funcional.

#### 4. Fluxo Básico de Eventos

O use case começa quando o motorista acede à função de atualização de dados.  
Insere o número de passageiros.  
O sistema guarda os dados e atualiza em tempo real.

#### 5. Fluxos Alternativos

##### **Erro na atualização**

O sistema solicita a correção dos dados.

#### 6. Fluxos Secundários

##### **Validar entrada de dados**

Confirmar o formato numérico válido.  
Guardar as informações.

#### 7. Cenários Principais

##### **Atualização bem-sucedida**

Motorista insere dados corretamente.  
O sistema atualiza em tempo real.

#### 8. Pós-Condições

A ocupação é atualizada no sistema.

#### 9. Requisitos Especiais

Os dados devem ser consistentes com a capacidade máxima do autocarro.

## 0.5 Gerir Bilhética

#### 1. Descrição

Permite que o passageiro solicite as funcionalidades relacionadas à bilhética, como comprar bilhetes, consultar saldos e histórico, etc.

#### 2. Atores

Passageiro	Solicita a gestão de bilhetes.
Aplicação	Processa as operações solicitadas.

### 3. Pré-Condições

O passageiro deve estar autenticado.

O sistema deve estar funcional.

### 4. Fluxo Básico de Eventos

O use case começa quando o passageiro seleciona uma opção na gestão da bilhética.

O sistema exibe as informações relacionadas à opção selecionada.

O passageiro completa as ações necessárias.

O sistema processa a solicitação e atualiza os dados.

### 5. Fluxos Alternativos

#### **Erro na execução**

Se a execução falhar, o sistema solicita uma nova tentativa.

### 6. Fluxos Secundários

#### **Validar a Saída de Dados**

Exibir ao utilizador apenas o pedido solicitado.

### 7. Cenários Principais

#### **Pedido bem sucedido**

O Passageiro solicita uma funcionalidade do sistema.

O sistema executa a funcionalidade.

### 8. Pós-Condições

O Passageiro recebe a informação relativa à funcionalidade solicitada.

### 9. Requisitos Especiais

Os dados devem ser criptografados.

## 0.6 Monitorizar o BRT

### 1. Descrição

Permite ao passageiro e ao administrador visualizar a localização e os horários dos autocarros em tempo real.

### 2. Atores

Passageiro	Consulta os horários e a localização.
Aplicação	Apresenta as informações.
Administrador	Supervisiona os veículos.

### 3. Pré-Condições

O sistema de localização e os sensores devem estar funcionais.

### 4. Basic Flow of Events

O utilizador acede à funcionalidade.  
A aplicação obtém os dados de GPS.  
Exibe as informações em tempo real.

### 5. Fluxos Alternativos

#### Erro no GPS ou sensores

O sistema informa a indisponibilidade temporária e solicita nova tentativa.

### 6. Fluxos Secundários

#### Recolher Dados do Sensor e GPS

Consultar localização atual.  
Verificar ocupação do autocarro.

### 7. Cenários Principais

#### Monitorizar a linha

O utilizador seleciona a linha.  
O sistema exibe os dados de localização e horários.

### 8. Pós-Condições

O utilizador recebe informações atualizadas.

## 9. Requisitos Especiais

A monitorização deve ser em tempo real.

## 0.7 Realizar Pagamento

### 1. Descrição

Permite ao passageiro efetuar pagamentos para creditar saldo ou comprar bilhetes.

### 2. Atores

Passageiro	Solicita o pagamento.
Aplicação	Processa o pagamento.

### 3. Pré-Condições

O passageiro deve ter acesso a métodos de pagamento válidos.

### 4. Fluxo Básico de Eventos

O use case começa quando o passageiro escolhe um bilhete ou creditar saldo.

O passageiro seleciona o método de pagamento.

O sistema processa o pagamento.

O sistema atualiza o saldo ou exibe o bilhete.

### 5. Fluxos Alternativos

#### Falha no pagamento

O sistema notifica o erro e solicita nova tentativa.

### 6. Fluxos Secundários

#### Validar pagamento

Confirmar os dados do pagamento.

Processar a transação.

### 7. Cenários Principais

#### Pagamento por qualquer método disponível

O passageiro escolhe o método.

O sistema processa e confirma o pagamento.

## 8. Pós-Condições

O pagamento é concluído com sucesso.

## 9. Requisitos Especiais

Todas as transações devem ser seguras.

# 0.8 Fazer Relatórios

## 1. Descrição

Permite que o administrador crie relatórios com base nos dados do sistema, como o uso dos bilhetes, ocupação e horários.

## 2. Atores

Administrador	Solicita os relatórios.
Aplicação	Cria e exibe os relatórios

## 3. Pré-Condições

Os dados devem estar disponíveis no sistema.

O administrador deve estar autenticado.

## 4. Fluxo Básico de Eventos

O use case começa quando o administrador solicita um relatório.

O sistema exibe as opções.

O administrador seleciona os filtros necessários.

O sistema gera e exibe o relatório.

## 5. Fluxos Alternativos

### Erro no processamento

O sistema notifica o erro e permite tentar novamente.

## 6. Fluxos Secundários

### Gerar relatório

O sistema processa os dados.

O sistema cria o relatório no formato solicitado.

## 7. Cenários Principais

### **Relatório do uso de bilhetes**

O administrador seleciona o período e a linha.

O sistema exibe o relatório.

### **Relatório de ocupação**

O administrador seleciona o período e a linha.

O sistema exibe o relatório.

### **Relatório de horários**

O administrador seleciona o período e a linha.

O sistema exibe o relatório.

## 8. Pós-Condições

O relatório é gerado e armazenado.

## 9. Requisitos Especiais

O sistema deve permitir exportar relatórios em diferentes formatos.

### **1.1 Registo na Aplicação**

#### 1. Descrição

O caso de uso "Registo na Aplicação" descreve as interações entre os utilizadores e o sistema para criar uma conta na aplicação.

#### 2. Atores

Passageiro	Pode criar uma conta, autenticar-se e atualizar os dados na aplicação.
Motorista	Pode criar uma conta, autenticar-se e atualizar os seus dados, incluindo detalhes relacionados às suas tarefas.
Administrador	Responsável pela gestão de contas dos utilizadores (criar, atualizar, desativar) e assegurar a manutenção do sistema.
Aplicação	Centraliza todas as interações e gere as funcionalidades relacionadas ao registo e à autenticação.

### 3. Pré-Condições

O utilizador deve ter acesso à aplicação.

### 4. Fluxo Básico de Eventos

O caso de uso começa quando um utilizador decide criar uma conta no sistema.  
O utilizador preenche as informações solicitadas.  
O sistema valida os dados fornecidos.  
O sistema executa a ação solicitada.  
O caso de uso termina com uma mensagem de sucesso ou de erro.

### 5. Fluxos alternativos

#### **Falha na validação dos dados**

Caso os dados fornecidos forem inválidos:

O sistema exibe uma mensagem de erro indicando qual é o problema.  
O utilizador é redirecionado para corrigir as informações.  
O caso de uso retoma o passo 2 do fluxo básico de eventos.

### 6. Fluxos Secundários

#### **Autenticação de Utilizador**

O utilizador insere as suas credenciais.  
O sistema verifica as credenciais fornecidas.  
O sistema autentica o utilizador e concede o acesso.

### 7. Cenários Principais

#### **Criação de Conta**

O utilizador preenche os dados solicitados (nome, email, palavra-passe).  
O sistema valida e armazena os dados.  
O utilizador recebe uma confirmação.

### 8. Pós-Condições

#### **Sucesso na identificação do utilizador e cruzamento de dados**

O utilizador está registado e autenticado.

#### **Fracasso na identificação do utilizador e cruzamento de dados**

O utilizador não é autenticado, podendo introduzir os dados outra vez.

### 9. Requisitos especiais

A aplicação deve possuir conexões seguras para proteger os dados dos utilizadores.

## 1.2 Autenticar o utilizador

### 1. Descrição

O utilizador pode autenticar-se no sistema utilizando as suas credenciais de acesso.

### 2. Atores

Passageiro	Autentica-se para aceder às funcionalidades disponíveis na aplicação.
Motorista	Autentica-se para gerir tarefas relacionadas às suas viagens.
Administrador	Autentica-se para aceder às permissões administrativas.

### 3. Pré-Condições

O utilizador já deve ter uma conta criada.

### 4. Fluxo Básico de Eventos

O caso de uso começa quando o utilizador insere o e-mail e a palavra-passe na página de login.

O sistema valida as credenciais fornecidas.

O sistema concede acesso às funcionalidades.

O caso de uso termina.

### 5. Fluxos Alternativos

#### Credenciais inválidas

Se as credenciais forem inválidas:

O sistema exibe uma mensagem de erro.

O utilizador pode tentar novamente.

### 6. Fluxos Secundários

#### Recuperação da Palavra-Passe

O utilizador solicita a recuperação da palavra-passe.

O sistema envia um email com instruções para redefinir a palavra-passe.

## 7. Cenários Principais

### **Autenticação bem sucedida**

O utilizador insere credenciais válidas e consegue aceder ao sistema.

## 8. Pós-Condições

O utilizador está autenticado no sistema.

## 9. Requisitos Especiais

A aplicação deve suportar uma autenticação segura.

## 1.3. Gerir as contas dos utilizadores

### 1. Descrição

O administrador pode gerir contas de utilizadores, como alterar permissões, bloquear contas ou excluí-las.

### 2. Atores

Administrador	Principal responsável pela gestão de contas na aplicação.
---------------	---

### 3. Pré-Condições

O administrador deve estar autenticado no sistema.

### 4. Fluxo Básico de Eventos

O caso de uso começa quando o administrador seleciona a opção "Gerir Contas".

O administrador pesquisa um utilizador específico.

O administrador escolhe uma ação (ex.: bloquear, alterar dados, etc.).

O sistema aplica as alterações.

O caso de uso termina.

### 5. Fluxos Alternativos

#### **Utilizador não encontrado**

Se o utilizador pesquisado não for encontrado:

O sistema exibe uma mensagem de erro.

O administrador pode realizar uma nova pesquisa.

## 6. Fluxos Secundários

### Bloquear uma Conta

O administrador seleciona a conta.

O sistema aplica o bloqueio e notifica o passageiro/motorista.

### Exclusão de uma Conta

O administrador seleciona uma conta.

O sistema aplica a exclusão e notifica o passageiro/motorista.

### Alteração de dados de uma Conta

O administrador seleciona uma conta.

O sistema aplica a alteração de dados e notifica o passageiro/motorista.

## 7. Cenários Principais

### Alterar permissões de utilizador

O administrador atualiza as permissões de uma conta específica.

## 8. Pós-Condições

As alterações são refletidas no sistema.

## 9. Requisitos Especiais

Apenas administradores autenticados podem realizar esta ação.

## 1.4 Atualizar os dados do utilizador

### 1. Descrição

Os utilizadores podem atualizar as suas informações.

### 2. Ator

Passageiro	Pode atualizar informações como e-mail ou palavra-passe.
Motorista	Pode atualizar informações relacionadas às suas tarefas.

### 3. Pré-Condições

O utilizador deve estar autenticado.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando o utilizador acessa o perfil e seleciona "Editar Dados".

O utilizador insere as informações a serem atualizadas.

O sistema valida os dados e aplica as alterações.

O sistema notifica o utilizador das alterações efetuadas.

O caso de uso termina.

#### 5. Fluxos Alternativos

##### Dados inválidos

Se os dados forem inválidos:

O sistema exibe uma mensagem de erro.

O utilizador pode tentar novamente.

#### 6. Fluxos Secundários

##### Atualizar palavra-passe

O utilizador insere a palavra-passe atual.

O sistema valida e solicita a nova palavra-passe.

##### Atualizar e-mail

O utilizador insere o e-mail atual.

O sistema valida e solicita o novo e-mail.

#### 7. Cenários Principais

##### Atualização de informações

O utilizador atualiza as informações com sucesso.

#### 8. Pós-Condições

Os dados do utilizador são atualizados no sistema.

#### 9. Requisitos Especiais

Dados como o e-mail devem ser validados para garantir a autenticidade.

### 1.5 Sair da aplicação

#### 1. Descrição

O utilizador encerra a sua sessão na aplicação.

## 2. Atores

Passageiro	Pode realizar logout para encerrar sua sessão.
Motorista	Pode realizar logout para encerrar sua sessão.
Administrador	Pode realizar logout para encerrar sua sessão.

## 3. Pré-Condições

O utilizador deve estar autenticado.

## 4. Fluxo Básico de Eventos

O caso de uso começa quando o utilizador seleciona a opção "Sair".

O sistema encerra a sessão do utilizador.

O utilizador é redirecionado para a tela inicial.

O caso de uso termina.

## 5. Fluxos Alternativos

N/A (o fluxo é direto e simples).

## 6. Fluxos Secundários

N/A

## 7. Cenários Principais

Encerrar sessão de forma segura

O sistema encerra a sessão e limpa os dados temporários.

## 8. Pós-Condições

A sessão do utilizador é encerrada.

## 9. Requisitos Especiais

O sistema deve assegurar que os dados temporários são excluídos após o logout.

## 2.1 Planear viagens conforme os passageiros

### 1. Descrição

O administrador planeia viagens com base nos dados dos passageiros tais como, destinos mais frequentes e horários preferidos.

### 2. Atores

Administrador	Responsável por analisar os dados de passageiros e planear rotas e horários.
Passageiro	Utilizador que fornece informações de uso indiretamente, como horários e locais preferidos de embarque/desembarque.
Aplicação	Ferramenta que recolhe os dados dos passageiros e ajuda na sugestão de itinerários.

### 3. Pré-Condições

O administrador deve estar autenticado no sistema.

Os dados dos passageiros devem estar disponíveis para análise.

### 4. Fluxo Básico de Eventos

O caso de uso começa quando o administrador seleciona a funcionalidade "Planear Viagens".

A aplicação exibe os dados recolhidos dos passageiros.

O administrador analisa os dados e define as rotas e horários otimizados.

O sistema guarda o plano e publica os horários para os passageiros e motoristas.

O caso de uso termina.

### 5. Fluxos Alternativos

#### Falta de dados dos passageiros

Se os dados forem insuficientes:

O sistema exibe uma mensagem a solicitar mais dados.

O administrador tenta novamente depois de recolher mais informações.

### 6. Fluxos Secundários

#### Publicação de rotas e horários

O sistema publica os horários definidos.

Os passageiros e motoristas são notificados sobre os novos itinerários.

## 7. Cenários Principais

### **Planeamento baseado na alta demanda**

O administrador ajusta os horários para atender horários de pico com maior demanda.

## 8. Pós-Condições

Os horários e rotas estão otimizados e disponíveis para serem usados.

## 9. Requisitos Especiais

O sistema deve permitir atualizações frequentes nos horários e rotas.

## 2.2 Registar o horário do BRT

### 1. Descrição

O administrador insere os horários do sistema BRT (Bus Rapid Transit) na aplicação.

### 2. Atores

Administrador	Responsável por registar os horários no sistema.
Aplicação	Processa e armazena os horários fornecidos pelo administrador.

### 3. Pré-condições

O administrador deve estar autenticado.

A aplicação deve estar disponível para edição de horários.

### 4. Fluxo Básico de Eventos

O caso de uso começa quando o administrador seleciona a funcionalidade "Registrar Horário".

O administrador insere os horários para cada rota.

O sistema valida os horários para evitar conflitos ou sobreposições.

O sistema guarda os horários e disponibiliza-os para os motoristas e passageiros.

O caso de uso termina.

## 5. Fluxos Alternativos

### **Conflito de horários**

Se houver sobreposição de horários:

O sistema exibe um aviso.

O administrador ajusta os horários e guarda novamente.

## 6. Fluxos Secundários

### **Publicação de horários atualizados**

O sistema publica os horários nas interfaces dos motoristas e passageiros.

## 7. Cenários Principais

### **Registo de um novo horário**

O administrador adiciona um horário para uma nova rota.

O sistema valida e publica a atualização.

## 8. Pós-Condições

Os horários do BRT estão atualizados no sistema.

## 9. Requisitos Especiais

O sistema deve evitar conflitos de horários automaticamente.

## 2.3 Informar os motoristas

### 1. Descrição

O sistema informa os motoristas sobre os horários, as rotas e as atualizações operacionais.

### 2. Atores

Administrador	Responsável por configurar as informações a serem enviadas.
Motorista	Recebe as informações relevantes para o seu trabalho.
Aplicação	Envia as informações configuradas aos motoristas.

### 3. Pré-Condições

Os motoristas devem estar registados no sistema.  
As informações dos horários e rotas devem estar atualizadas.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando o administrador atualiza ou publica informações sobre as rotas.

O sistema processa as informações e envia-as para os motoristas.  
Os motoristas recebem as atualizações nos seus dispositivos.  
O caso de uso termina.

#### 5. Fluxos Alternativos

N/A (fluxo direto).

#### 6. Fluxos Secundários

##### **Receber informações operacionais**

O motorista recebe notificações com os horários e alterações de rotas.

#### 7. Cenários Principais

##### **Enviar notificações de alteração de rota**

O administrador ajusta uma rota.  
O sistema informa imediatamente os motoristas afetados pelo ajuste.

#### 8. Pós-Condições

Os motoristas estão informados sobre as suas rotas e horários.

#### 9. Requisitos Especiais

O sistema deve suportar notificações em tempo real.

### **2.4 Registar os motoristas responsáveis pelo transporte**

#### 1. Descrição

O administrador regista os motoristas no sistema e associa-os a rotas específicas.

#### 2. Atores

Administrador	Responsável por adicionar motoristas ao sistema.
---------------	--

Aplicação	Armazena as informações dos motoristas e associa-os às rotas.
-----------	---

### 3. Pré-Condições

O administrador deve estar autenticado.  
Os motoristas devem estar registados no sistema.

### 4. Fluxo Básico de Eventos

O caso de uso começa quando o administrador seleciona a funcionalidade "Registrar Motoristas".  
O administrador insere os dados do motorista.  
O sistema valida e armazena as informações.  
O sistema associa o motorista às rotas específicas.  
O caso de uso termina.

### 5. Fluxos Alternativos

#### Dados incompletos

Se os dados do motorista estiverem incompletos:  
O sistema notifica o administrador.  
O administrador completa os dados antes de guardar.

### 6. Fluxos Secundários

#### Atualização de motoristas às rotas

O administrador pode alterar as associações de motoristas às rotas.

### 7. Cenários Principais

#### Adicionar um novo motorista

O administrador insere os dados do motorista e associa-os a uma rota.

### 8. Pós-Condições

O motorista está registrado e associado a uma rota específica.

### 9. Requisitos Especiais

O sistema deve verificar se o motorista já está associado a outra rota no mesmo horário.

## 2.5 Recolher dados em tempo real

### 1. Descrição

O sistema recolhe informações em tempo real sobre a operação do transporte, bem como a localização dos veículos e a sua ocupação.

### 2. Atores

Administrador	Pode visualizar os dados em tempo real para monitorar o sistema.
Aplicação	Recolhe e organiza os dados operacionais.

### 3. Pré-Condições

O sistema deve estar conectado aos dispositivos de monitoramento.

### 4. Fluxos Básicos de Eventos

O caso de uso começa quando o sistema inicia a recolha de dados em tempo real. O sistema monitora e armazena informações, como a localização dos veículos e a sua ocupação.

O administrador pode visualizar os dados em tempo real na interface do sistema. O caso de uso termina.

### 5. Fluxos Alternativos

#### Falha na recolha de dados

O sistema informa a falha e tenta restabelecer a conexão automaticamente.

### 6. Fluxos Secundários

#### Visualização de dados em tempo real

O administrador acede aos dados de localização e ocupação no painel de controlo.

### 7. Cenários Principais

#### Monitorização da ocupação

O sistema alerta quando a ocupação de um veículo excede o limite.

#### Monitorização da localização

O sistema alerta quando a localização do veículo é perdida.

## 8. Pós-Condições

Os dados são armazenados para análises posteriores.

## 9. Requisitos Especiais

O sistema deve suportar a integração de outros dispositivos para monitoramento.

### 3.1 Validar o bilhete

#### 1. Descrição

Este caso de uso descreve o processo pelo qual o passageiro valida o seu bilhete no sistema. A aplicação verifica a validade do bilhete, e o passageiro pode ser autorizado a entrar no autocarro se o bilhete for válido.

#### 2. Atores

Passageiro	Utilizador do sistema que deseja validar o bilhete.
Aplicação	Sistema utilizado para validar o bilhete.

#### 3. Pré-condições

O passageiro possui um bilhete válido (físico ou digital).

O sistema de validação está em funcionamento e acessível.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando o passageiro apresenta o bilhete à aplicação.

A aplicação lê os dados do bilhete e realiza a validação.

Se o bilhete for válido, o sistema confirma a validação.

O passageiro recebe confirmação de que pode entrar no autocarro.

O caso de uso termina com a validação bem-sucedida do bilhete.

#### 5. Fluxos Alternativos

##### Bilhete inválido

Se o bilhete for inválido:

A aplicação exibe uma mensagem de erro.

O fiscalizador é notificado para verificar a situação.

O caso de uso termina com um erro de validação.

## 6. Fluxos Secundários

A aplicação verifica se o bilhete corresponde a um bilhete válido na base de dados. Se o bilhete não for encontrado, o sistema indica que a validação falhou.

## 7. Cenários Principais

O passageiro valida o bilhete corretamente.

A aplicação confirma a validação e permite o embarque.

## 8. Pós-condições

O bilhete foi validado com sucesso e o passageiro pode acessar o transporte.

## 9. Requisitos Especiais

O sistema de validação deve ser rápido e garantir a precisão na leitura dos bilhetes.

# 3.2 Verificar o Bilhete

## 1. Descrição

Este caso de uso descreve o processo em que o fiscalizador verifica o bilhete do passageiro após a tentativa de validação. O objetivo é confirmar se o bilhete é válido e se o passageiro tem direito de realizar a viagem.

## 2. Ator

Fiscalizador	Responsável por verificar manualmente a validade do bilhete.
--------------	--

## 3. Pré-condições

O passageiro tentou validar o bilhete.

O fiscalizador tem acesso ao sistema para verificar a situação do bilhete.

## 4. Fluxo Básico de Eventos

O caso de uso começa quando o fiscalizador consulta o estado da validação do bilhete na aplicação.

A aplicação exibe as informações do bilhete.

O fiscalizador verifica se o bilhete está válido.

Se o bilhete for válido, o fiscalizador confirma a permissão de embarque.

O caso de uso termina com a confirmação da validade do bilhete.

## 5. Fluxos Alternativos

### Bilhete não encontrado

Se o bilhete não for encontrado ou for inválido:

O fiscalizador solicita um novo bilhete ou outras informações ao passageiro.

O fiscalizador regista o incidente no sistema.

O caso de uso termina com uma falha de verificação.

## 6. Fluxos Secundários

O fiscalizador consulta o histórico de validações passadas do passageiro na aplicação.

Se houver problemas recorrentes, o fiscalizador procede a uma ação de correção.

## 7. Cenários Principais

O fiscalizador valida que o bilhete está correto.

O passageiro é autorizado a embarcar.

## 8. Pós-condições

O bilhete foi verificado e o fiscalizador tomou uma decisão com base na sua validade.

## 9. Requisitos Especiais

A aplicação deve ter uma interface clara para que o fiscalizador consulte rapidamente os detalhes do bilhete.

## 3.3 Atualizar a ocupação do autocarro

### 1. Descrição

Este caso de uso descreve o processo em que o fiscalizador atualiza a ocupação do autocarro após cada validação de bilhete. A aplicação regista o número de passageiros a bordo.

### 2. Ator

Aplicação	Sistema que atualiza a ocupação.
-----------	----------------------------------

### 3. Pré-condições

O fiscalizador tem acesso ao sistema de ocupação.

O passageiro validou o bilhete.

#### 4. Fluxo Básico de Eventos

A aplicação atualiza a ocupação do autocarro assim que um passageiro valida o seu bilhete.

O fiscalizador pode visualizar a ocupação atualizada na aplicação.

O caso de uso termina com a ocupação do autocarro atualizada.

#### 5. Fluxos Alternativos

##### **Superar limite de ocupação**

Se a ocupação ultrapassar o limite:

A aplicação avisa o fiscalizador.

O fiscalizador pode decidir não permitir mais passageiros.

#### 6. Fluxos Secundários

A aplicação verifica a capacidade do autocarro antes de permitir mais passageiros.

#### 7. Cenários Principais

O passageiro tem sua entrada registada na aplicação.

A aplicação atualiza a ocupação do autocarro corretamente.

#### 8. Pós-condições

A ocupação do autocarro foi atualizada corretamente.

#### 9. Requisitos Especiais

O sistema deve permitir a atualização em tempo real e ser eficiente.

### 3.4 Reportar problemas de validação

#### 1. Descrição

Este caso de uso descreve como o fiscalizador reporta problemas relacionados à validação do bilhete para a aplicação.

#### 2. Atores

Fiscalizador	Responsável por reportar os problemas.
--------------	--

Aplicação	O sistema que regista os problemas e os envia para análise.
-----------	---

### 3. Pré-condições

O fiscalizador identificou um problema durante a validação do bilhete.  
O fiscalizador tem acesso ao sistema da aplicação para reportar problemas.  
A aplicação está operacional para receber relatórios de falha.  
O fiscalizador tem permissão para realizar o reporte de problemas no sistema.

### 4. Fluxo Básico de Eventos

O fiscalizador identifica que houve um erro durante a tentativa de validação do bilhete.  
O fiscalizador regista o problema através da aplicação.  
A aplicação recolhe detalhes do problema, como o tipo de erro, a hora do incidente, e as informações do bilhete.  
O sistema valida que todas as informações necessárias foram fornecidas antes de permitir o envio.  
A aplicação envia os dados do problema para o administrador.  
O caso de uso termina com a notificação do administrador sobre o problema.

### 5. Fluxos Alternativos

#### Falha no sistema de reporte

Se houver uma falha no sistema de reporte, como uma desconexão ou erro técnico:  
O fiscalizador é notificado da falha e é solicitado a tentar novamente.  
O fiscalizador tenta novamente submeter o problema.  
Se a falha persistir, o fiscalizador pode registar o incidente manualmente ou aguardar a correção do sistema.

### 6. Fluxos Secundários

Após o envio do relatório de erro, a aplicação notifica o administrador para que ele possa iniciar o processo de resolução.  
O administrador recebe uma notificação do tipo de problema e toma as providências necessárias, como investigar a causa do erro.

### 7. Cenários Principais

O fiscalizador reporta corretamente o problema no sistema.  
A aplicação regista e envia os dados para o administrador.  
O administrador é notificado para iniciar a resolução.

## 8. Pós-condições

O problema foi reportado corretamente, e o administrador recebeu as informações para investigar a falha.

## 9. Requisitos Especiais

O processo de reporte de falhas deve ser transparente e fácil de utilizar, sem sobrecarregar o fiscalizador com etapas complexas.

### 3.5 Consultar o histórico de validações

#### 1. Descrição

Este caso de uso descreve como o administrador consulta o histórico de validações de bilhetes para monitorizar e analisar as operações de validação.

#### 2. Ator

Administrador	Responsável por aceder e analisar o histórico de validações de bilhetes.
---------------	--

#### 3. Pré-condições

O administrador tem acesso ao sistema da aplicação.

O histórico de validações está armazenado e disponível na aplicação.

O administrador tem permissões adequadas para consultar o histórico.

#### 4. Fluxo Básico de Eventos

O administrador acede à aplicação para consultar o histórico de validações de bilhetes.

A aplicação exibe o histórico de validações, podendo ser filtrado por data, tipo de bilhete ou passageiro.

O administrador analisa o histórico para identificar padrões ou problemas relacionados com a validação de bilhetes.

O caso de uso termina quando o administrador termina a consulta e analisa as informações.

#### 5. Fluxos Alternativos

##### Histórico incompleto

Se o histórico não contiver todos os registos esperados:

O administrador tenta obter os dados em falta.

Caso a falha persista, o administrador pode reportar o problema ao suporte técnico.

#### 6. Fluxos Secundários

O administrador aplica filtros para polir a pesquisa, como por data ou tipo de bilhete.

A aplicação exibe os resultados filtrados conforme os critérios definidos.

#### 7. Cenários Principais

O administrador acede ao histórico de validações.

A aplicação exibe os dados conforme esperado, permitindo ao administrador analisar as informações.

#### 8. Pós-condições

O administrador consultou o histórico de validações com sucesso.

#### 9. Requisitos Especiais

A aplicação deve garantir que os dados sejam apresentados de forma rápida e precisa, mesmo com grandes volumes de informações.

### 4.1 Registar entrada de Passageiros

#### 1. Descrição

Este caso de uso descreve o processo no qual o Sistema de Câmaras regista a entrada de passageiros no autocarro, monitorizando a ocupação.

#### 2. Ator

Sistema de Câmaras	Regista a entrada de passageiros através da câmara de monitorização.
--------------------	--

#### 3. Pré-condições

O Sistema de Câmaras está operacional e a monitorizar a entrada do autocarro.  
O passageiro está a entrar no autocarro.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando o Sistema de Câmaras deteta a entrada do passageiro no autocarro.

O Sistema de Câmaras regista a entrada na base de dados.

O caso de uso termina com o Sistema de Câmaras tendo registado a entrada.

## 5. Fluxos Alternativos

### Falha na detecção

Se o Sistema de Câmaras falhar ao detetar a entrada:

O sistema tenta novamente detectar a entrada.

Se a falha persistir, o fiscalizador é notificado.

## 6. Fluxos Secundários

O Sistema de Câmaras analisa a imagem capturada.

O sistema regista a entrada no banco de dados, associando ao passageiro.

## 7. Cenários Principais

O Sistema de Câmaras detecta e regista corretamente a entrada do passageiro.

A entrada é armazenada na base de dados.

## 8. Pós-condições

A entrada do passageiro foi registada com sucesso.

## 9. Requisitos Especiais

O Sistema de Câmaras deve ser capaz de funcionar em tempo real, com alta precisão na detecção.

## 4.2 Registar entrada de Passageiros

### 1. Descrição

Este caso de uso descreve o processo no qual o **Sistema de Câmeras** regista a saída de passageiros do autocarro.

### 2. Ator

Sistema de Câmaras	Regista a saída de passageiros através da câmara de monitorização.
--------------------	--

### 3. Pré-condições

O Sistema de Câmaras está operacional e a monitorizar a saída do autocarro.  
O passageiro está a sair do autocarro.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando o Sistema de Câmaras deteta a entrada do passageiro no autocarro.

O Sistema de Câmaras regista a entrada na base de dados.

O Sistema de Câmaras valida a entrada, confirmado que o registo foi feito corretamente.

O caso de uso termina com o Sistema de Câmaras tendo registado a entrada.

#### 5. Fluxos Alternativos

##### **Falha na detecção**

Se o Sistema de Câmaras falhar ao detetar a saída:

O sistema tenta novamente detectar a saída.

Se persistir a falha, o fiscalizador é alertado.

#### 6. Fluxos Secundários

O Sistema de Câmaras analisa a imagem capturada.

O sistema regista a saída na base de dados.

#### 7. Cenários Principais

O Sistema de Câmaras detecta e regista corretamente a saída do passageiro.  
A saída é armazenada na base de dados.

#### 8. Pós-condições

A saída do passageiro foi registada com sucesso.

#### 9. Requisitos Especiais

O Sistema de Câmaras deve garantir uma alta taxa de deteção para minimizar erros.

### 4.3 Atualizar os dados na aplicação

#### 1. Descrição

Este caso de uso descreve o processo no qual o Sistema de Câmaras e a Aplicação atualizam os dados da ocupação na aplicação, para garantir que as informações são atualizadas em tempo real.

## 2. Atores

Sistema de Câmaras	Regista os dados de entrada e saída dos passageiros.
Aplicação	Atualiza os dados da ocupação do autocarro na interface de utilizador.

## 3. Pré-condições

O Sistema de Câmaras registou as entradas e saídas dos passageiros.  
A Aplicação está pronta para receber os dados e atualizá-los.

## 4. Fluxo Básico de Eventos

O caso de uso começa quando o Sistema de Câmaras envia os dados de entrada e de saída para a aplicação.  
A aplicação atualiza a ocupação do autocarro com os dados recebidos.  
A aplicação valida a atualização dos dados para garantir que estão correctos.  
O caso de uso termina com a aplicação actualizada.

## 5. Fluxos Alternativos

### Dados incompletos

Se a aplicação receber dados incompletos ou inválidos:  
A aplicação solicita novos dados ao Sistema de Câmaras.  
O fiscalizador é notificado para intervir, se necessário.

## 6. Fluxos Secundários

A aplicação utiliza os dados de entrada e saída para calcular a ocupação.  
A aplicação mostra a ocupação do autocarro na interface do utilizador.

## 7. Cenários Principais

A aplicação recebe os dados corretamente e atualiza a ocupação.

## 8. Pós-condições

A ocupação do autocarro foi atualizada na aplicação.

## 9. Requisitos Especiais

A aplicação deve atualizar os dados em tempo real para refletir a ocupação atual.

## 4.4 Calcular a ocupação actual

### 1. Descrição

Este caso de uso descreve como a aplicação calcula a ocupação atual do autocarro com base nos dados de entrada e saída dos passageiros.

### 2. Ator

Aplicação	Responsável por calcular a ocupação actual do autocarro
-----------	---

### 3. Pré-condições

A aplicação recebeu dados actualizados sobre entradas e saídas.  
A aplicação tem acesso à capacidade máxima do autocarro.

### 4. Fluxo Básico de Eventos

O caso de uso começa quando a aplicação recebe os dados actualizados da ocupação.

A aplicação calcula a ocupação atual, comparando entradas e saídas.

A aplicação verifica se a ocupação calculada excede a capacidade do autocarro.

O caso de uso termina quando a aplicação calcula e armazena a ocupação atual.

### 5. Fluxos Alternativos

#### Falha no cálculo

Se houver erro no cálculo da ocupação:

A aplicação alerta o fiscalizador.

O fiscalizador pode verificar manualmente a ocupação.

### 6. Fluxos Secundários

A aplicação compara a ocupação calculada com a capacidade máxima do autocarro.

Se a capacidade for excedida, a aplicação alerta o fiscalizador.

### 7. Cenários Principais

A aplicação calcula e exibe corretamente a ocupação.

## 8. Pós-condições

A ocupação foi calculada correctamente.

## 9. Requisitos Especiais

A aplicação deve ser rápida e precisa no cálculo da ocupação.

# 4.5 Disponibilizar dados da ocupação

## 1. Descrição

Este caso de uso descreve como a aplicação disponibiliza os dados da ocupação para o fiscalizador ou outros sistemas.

## 2. Ator

Aplicação	Responsável por disponibilizar os dados de ocupação.
-----------	--

## 3. Pré-condições

A aplicação tem os dados da ocupação calculados.

## 4. Fluxo Básico de Eventos

A aplicação disponibiliza os dados da ocupação para os utilizadores autorizados. O fiscalizador ou outros sistemas acedem os dados da aplicação.

A aplicação atualiza os dados em tempo real, garantindo que as informações disponibilizadas são sempre as mais recentes.

O fiscalizador ou o sistema acede aos dados e toma as ações necessárias com base nas informações disponibilizadas.

O caso de uso termina com a disponibilização bem-sucedida dos dados.

## 5. Fluxos Alternativos

### Falha na disponibilização dos dados

Se a aplicação falhar ao disponibilizar os dados:

O fiscalizador tenta aceder aos dados novamente.

Se o erro persistir, o fiscalizador reportará o problema ao suporte técnico.

## 6. Fluxos Secundários

O fiscalizador pode aplicar filtros, como data ou tipo de ocupação.

A aplicação exibe os resultados filtrados conforme os critérios do fiscalizador.

#### 7. Cenários Principais

A aplicação disponibiliza os dados da ocupação corretamente.

O fiscalizador ou o sistema consulta e utiliza os dados.

#### 8. Pós-condições

Os dados da ocupação foram disponibilizados com sucesso e estão acessíveis para os utilizadores autorizados.

#### 9. Requisitos Especiais

A aplicação deve ser capaz de disponibilizar dados em tempo real, com alta disponibilidade e precisão.

### 4.6 Relatar problemas na contagem

#### 1. Descrição

Este caso de uso descreve o processo no qual o motorista reporta problemas com a contagem ou monitorização da ocupação do autocarro para a aplicação.

#### 2. Ator

Motorista	Responsável por reportar problemas relacionados com a contagem ou monitorização da ocupação do autocarro.
-----------	---

#### 3. Pré-condições

O motorista identificou um problema com a contagem ou ocupação do autocarro.

O motorista tem acesso à aplicação para reportar o problema.

A aplicação está operacional para receber os relatórios de falhas.

#### 4. Fluxo Básico de Eventos

O motorista detecta um problema relacionado com a ocupação ou contagem de passageiros no autocarro.

O motorista reporta o problema através da aplicação.

A aplicação regista o problema, recolhendo os detalhes necessários (ex: tipo de erro, hora do incidente, etc.).

O caso de uso termina com o relatório bem-sucedido do problema para a aplicação.

## 5. Fluxos Alternativos

### **Falha na transmissão do relatório**

Se houver uma falha na transmissão do relatório:

O motorista tenta novamente submeter o problema.

Se a falha persistir, o motorista pode registar o problema manualmente ou aguardar a resolução do erro técnico.

## 6. Fluxos Secundários

Após o relatório, a aplicação envia os dados do problema para o administrador. O administrador toma as ações necessárias para resolver o problema.

## 7. Cenários Principais

O motorista reporta o problema corretamente.

A aplicação regista o problema e envia as informações para o administrador

## 8. Pós-condições

O problema foi reportado e o administrador foi notificado para iniciar a resolução.

## 9. Requisitos Especiais

O sistema de reporte deve ser eficiente, garantindo que os problemas são registados de forma clara e precisa, permitindo ações rápidas.

## 5.1 Recarregar Cartão

### 1. Descrição

O passageiro carrega o saldo do seu cartão utilizando diferentes métodos de pagamento (MB Way, PayPal ou dinheiro físico).

### 2. Atores

Passageiro	Inicia o processo de recarga.
Aplicação	Processa a transação e atualiza o saldo.

### 3. Pré-Condições

O passageiro deve estar autenticado na aplicação.

O sistema de pagamento deve estar operacional.

#### 4. Fluxo Básico de Eventos

O passageiro acessa a opção "Recarregar Cartão" na aplicação.

Escolhe o valor e o método de pagamento.

A aplicação processa o pagamento e atualiza o saldo do cartão.

O saldo atualizado é exibido ao passageiro.

#### 5. Fluxos Alternativos

##### **Falha no pagamento**

A aplicação informa o erro e solicita a repetição do processo.

O caso de uso retoma no passo 2 do fluxo básico.

#### 6. Fluxos Secundários

##### **Validar Pagamento**

O sistema verifica os dados do pagamento.

Processa ou rejeita a transação.

#### 7. Cenários Principais

##### **Recarregar via MB Way**

O passageiro escolhe a opção MB Way, insere o valor e finaliza a transação.

#### 8. Pós-Condições

O saldo do cartão é atualizado com sucesso.

#### 9. Requisitos Especiais

As transações devem ser encriptadas.

## 5.2 Comprar Bilhete

#### 1. Descrição

Permite ao passageiro comprar bilhetes para utilização no sistema de transporte.

#### 2. Atores

Passageiro	Realiza a compra de bilhetes.
------------	-------------------------------

Aplicação	Gera e armazena os bilhetes adquiridos.
-----------	---

### 3. Pré-Condições

O passageiro deve estar autenticado.  
Deve haver saldo suficiente para realizar a compra.

### 4. Fluxo Básico de Eventos

O passageiro seleciona o tipo de bilhete na aplicação.  
O sistema verifica o saldo ou processa o pagamento.  
O bilhete é gerado e exibido na aplicação.

### 5. Fluxo Alternativo

#### **Saldo insuficiente**

O sistema solicita a recarga do cartão.  
O caso de uso retoma no passo 2 do fluxo básico.

### 6. Fluxos Secundários

#### **Validar Dados de Compra**

O sistema verifica o bilhete solicitado.  
O sistema processa a transação.

### 7. Cenários Principais

#### **Compra de bilhete mensal**

O passageiro compra um bilhete mensal e o sistema regista-o na aplicação.

### 8. Pós-Condições

O bilhete é gerado e armazenado na conta do passageiro.

### 9. Requisitos Especiais

O sistema deve suportar vários métodos de pagamento para compra de bilhetes.

## 5.3 Consultar/Validar Bilhete

### 1. Descrição

Permite a verificação da validade dos bilhetes.

### 2. Atores

Passageiro	Verifica se o bilhete está válido.
Fiscalizador	Valida os bilhetes durante as viagens.
Aplicação	Processa e verifica o status do bilhete.

### 3. Pré-Condições

O bilhete deve ter sido previamente gerado.

### 4. Fluxo Básico de Eventos

O bilhete é consultado no sistema.

A aplicação verifica a validade na base de dados.

O sistema devolve o estado do bilhete.

### 5. Fluxos Alternativos

#### Bilhete expirado ou inválido:

O sistema exibe "inválido" ou "expirado".

### 6. Fluxos Secundários

#### Consulta do estado

A aplicação verifica a validade e a data de expiração do bilhete.

### 7. Cenários Principais

#### Fiscalizador valida o bilhete

O fiscalizador verifica a validade do bilhete.

### 8. Pós-Condições

O estado do bilhete é exibido ao utilizador ou fiscalizador.

## 9. Requisitos Especiais

A validação deve ser rápida e confiável.

## 5.4 Cancelar Bilhete

### 1. Descrição

Permite ao passageiro cancelar um bilhete previamente adquirido.

### 2. Atores

Passageiro	Solicita o cancelamento.
Aplicação	Processa a solicitação e atualiza o estado do bilhete.

### 3. Pré-Condições

O bilhete deve estar dentro do prazo para cancelamento.

### 4. Fluxo Básico de Eventos

O passageiro solicita o cancelamento do bilhete na aplicação.  
A aplicação verifica se o bilhete é elegível para cancelamento.  
O bilhete é cancelado e o saldo (ou valor) é reembolsado.

### 5. Fluxos Alternativos

#### Bilhete não cancelável

O sistema informa que o bilhete não pode ser cancelado.  
O valor não é reembolsado.

### 6. Fluxos Secundários

#### Processar o Reembolso

O sistema calcula o valor a ser reembolsado e atualiza o saldo.

### 7. Cenários Principais

#### Cancelamento de bilhete válido

O passageiro cancela o bilhete antes do uso e recebe reembolso.

### 8. Pós-Condições

O bilhete é cancelado e removido do sistema.

#### 9. Requisitos Especiais

As regras de cancelamento devem ser claras e aplicáveis.

### 5.5 Reportar Problemas no Sistema

#### 1. Descrição

Permite ao utilizador reportar problemas técnicos ou relacionados à bilhética.

#### 2. Atores

Passageiro	Reportam problemas identificados.
Fiscalizador	Reportam problemas identificados.
Aplicação	Registra e encaminha os problemas ao suporte.

#### 3. Pré-Condições

O utilizador deve estar autenticado na aplicação.

#### 4. Fluxo Básico de Eventos

O utilizador escolhe a opção "Reportar Problemas".

Descreve o problema.

A aplicação regista o problema e envia uma confirmação.

#### 5. Fluxos Alternativos

N/A (fluxo direto e simples).

#### 6. Fluxos Secundários

##### Encaminhar o Problema ao Suporte

O sistema envia os detalhes do problema para a equipa de suporte.

## 7. Cenários Principais

### **Problema técnico relatado pelo fiscalizador**

O fiscalizador relata algum problema.

### **Problema técnico relatado pelo utilizador**

O utilizador relata algum problema.

## 8. Pós-Condições

O problema é registrado no sistema para análise posterior.

## 9. Requisitos Especiais

A aplicação deve armazenar o histórico dos problemas relatados.

## 5.6 Fazer Relatórios sobre a Bilhética

### 1. Descrição

Permite ao administrador gerar relatórios operacionais e financeiros relacionados à bilhética.

### 2. Atores

Administrador	Gera relatórios para análise.
Aplicação	Consolida e apresenta os dados.

### 3. Pré-Condições

O sistema deve ter os dados atualizados sobre os bilhetes e as transações.

### 4. Fluxo Básico de Eventos

O administrador abre a opção de "Fazer Relatórios".

Escolhe o tipo de relatório desejado (financeiro ou operacional).

A aplicação processa os dados e gera o relatório.

O relatório é exibido ou exportado.

### 5. Fluxos Alternativos

N/A (fluxo direto e simples).

## 6. Fluxos Secundários

### **Exportar Relatório**

O relatório é gerado e guardado em PDF ou Excel.

### **Exibir Relatório**

O relatório é gerado e exibido.

## 7. Cenários Principais

### **Relatório financeiro gerado pelo administrador**

O administrador analisa o relatório financeiro mensal.

### **Relatório operacional gerado pelo administrador**

O administrador analisa o relatório operacional mensal.

## 8. Pós-Condições

O relatório é gerado e fica disponível para o administrador.

## 9. Requisitos Especiais

O sistema deve suportar a exportação dos relatórios em múltiplos formatos.

## 6.1 Monitorizar Ocupação

### 1. Descrição

Permite monitorizar, em tempo real, o nível de ocupação dos autocarros.

### 2. Atores

Administrador	Consulta as informações de ocupação para avaliar a eficiência do sistema.
Passageiro	Verifica a ocupação dos autocarros antes de decidir embarcar.
Sensor	Envia os dados de ocupação dos autocarros ao sistema.
Aplicação	Processa os dados recebidos e exibe informações aos utilizadores.

### 3. Pré-Condições

Os sensores de ocupação devem estar sempre operacionais.  
A aplicação tem de estar conectada ao sistema.

### 4. Fluxo Básico de Eventos

Os sensores instalados nos autocarros transmitem dados de ocupação ao sistema.  
A aplicação processa e atualiza as informações em tempo real.  
O administrador ou passageiro acede à aplicação e consulta o nível de ocupação.  
A informação é exibida ao utilizador.

### 5. Fluxos Alternativos

#### **Sensor inativo**

A aplicação exibe uma mensagem de indisponibilidade de dados.

#### **Sensor sem conexão**

A aplicação exibe uma mensagem de indisponibilidade de dados.

### 6. Fluxos Secundários

#### **Processamento dos Dados de Ocupação**

Os dados são enviados pelos sensores e processados pela aplicação antes de serem exibidos.

### 7. Cenários Principais

#### **Monitorização em tempo real**

O administrador confere a ocupação durante horários de pico.  
O passageiro confere a ocupação quando decide usufruir do transporte.

### 8. Pós-Condições

Os dados de ocupação têm de ser sempre atualizados em tempo real.

### 9. Requisitos Especiais

Os sensores devem enviar dados de forma contínua e sem interrupções.

## 6.2 Consultar a Localização

### 1. Descrição

Permite verificar a localização em tempo real dos autocarros.

### 2. Atores

Administrador	Consulta a localização atual dos autocarros.
Passageiro	Consulta a localização atual dos autocarros.
Sistema de Localização	Fornece as coordenadas dos autocarros.
Aplicação	Exibe a localização dos autocarros num mapa.

### 3. Pré-Condições

Os autocarros devem estar equipados com dispositivos de GPS funcionais.  
A aplicação deve ter acesso ao sistema de localização.

### 4. Fluxo Básico de Eventos

Os dispositivos de GPS enviam a localização atual dos autocarros ao sistema de localização.  
O sistema de localização passa os dados para a aplicação.  
O administrador ou passageiro acede à funcionalidade de localização na aplicação.  
A localização dos autocarros é exibida na interface.

### 5. Fluxos Alternativos

#### Falta de conexão com o GPS

A aplicação informa a indisponibilidade de dados de localização.

### 6. Fluxos Secundários

#### Atualização dos Dados de Localização

O sistema de localização envia atualizações periódicas para a aplicação.

### 7. Cenários Principais

#### Passageiro consulta a localização

O passageiro verifica a localização do autocarro antes de se dirigir à paragem.

#### **Administrador consulta a localização**

O administrador verifica a localização do autocarro para confirmar que está tudo dentro da normalidade.

#### [8. Pós-Condições](#)

A localização dos autocarros é exibida corretamente e em tempo real.

#### [9. Requisitos Especiais](#)

O sistema deve permitir a exibição da localização em formato de mapa.

### **6.3 Obter Dados do BRT**

#### [1. Descrição](#)

Fornece informações consolidadas sobre o BRT, incluindo a ocupação, a localização e os horários.

#### [2. Atores](#)

Administrador	Consulta os dados para futura análise e tomada de decisões.
Passageiro	Obtém informações relevantes para planear a sua viagem.
Aplicação	Processa e consolida os dados do sistema.

#### [3. Pré-Condições](#)

O sistema de sensores e de localização devem estar funcionais.

A aplicação deve estar a operar.

#### [4. Fluxo Básico de Eventos](#)

O administrador ou passageiro solicita os dados consolidados na aplicação.

A aplicação processa as informações de várias fontes (sensores, GPS).

Os dados consolidados são exibidos ao utilizador.

## 5. Fluxos Alternativos

N/A (fluxo direto e simples).

## 6. Fluxos Secundários

### Consolidar Dados

A aplicação reúne informações dos sensores e do sistema de localização.

## 7. Cenários Principais

### O Administrador consulta os dados em hora de pico

O administrador verifica a ocupação e os horários durante períodos de alta demanda.

### O Passageiro consulta os dados

O passageiro verifica a ocupação e os horários para planear a sua viagem.

## 8. Pós-Condições

Os dados do sistema BRT são apresentados ao utilizador de forma consolidada.

## 9. Requisitos Especiais

O sistema deve processar grandes volumes de dados de forma rápida e eficiente.

## 6.4 Estimar Tempo de Chegada

### 1. Descrição

Calcula o tempo estimado para um autocarro chegar a uma paragem específica.

### 2. Atores

Administrador	Consulta o tempo estimado de chegada dos autocarros.
Passageiro	Consulta o tempo estimado de chegada dos autocarros.
Sistema de Localização	Fornece a posição dos autocarros.
Aplicação	Calcula o tempo com base na localização, trânsito, etc.

### 3. Pré-Condições

O sistema de localização e a base de dados dos horários devem estar operacionais.

### 4. Fluxo Básico de Eventos

O utilizador solicita o tempo estimado de chegada de um autocarro.

O sistema de localização fornece a posição atual do autocarro.

A aplicação calcula o tempo estimado com base nos dados fornecidos.

O tempo estimado é exibido ao utilizador.

### 5. Fluxos Alternativos

#### **Falta de conexão com o sistema de localização**

A aplicação exibe uma estimativa baseada nos horários padrão.

### 6. Fluxos Secundários

#### **Cálculo do Tempo Estimado**

A aplicação utiliza dados de localização e condições de tráfego para calcular o tempo.

### 7. Cenários Principais

#### **Passageiro consulta o tempo estimado de chegada**

O passageiro verifica quanto tempo falta para o autocarro chegar à sua paragem.

#### **Administrador consulta o tempo estimado**

O administrador verifica quanto tempo falta para o autocarro chegar à próxima paragem de modo a confirmar que está tudo operacional.

### 8. Pós-Condições

O tempo estimado de chegada é apresentado a quem o solicitar.

### 9. Requisitos Especiais

O sistema deve integrar dados de tráfego em tempo real para uma maior precisão.

## **6.5 Exibir as Informações na Interface da Aplicação**

### 1. Descrição

Apresenta as informações do sistema BRT na interface da aplicação.

## 2. Atores

Administrador	Visualiza as informações consolidadas na aplicação.
Passageiro	Visualiza as informações consolidadas na aplicação.
Aplicação	Processa e exibe os dados de forma organizada.

## 3. Pré-Condições

O sistema deve estar funcional e conectado às fontes dos dados.

## 4. Fluxo Básico de Eventos

A aplicação recebe as informações consolidadas do sistema.

Os dados são organizados e exibidos na interface.

O utilizador acede às informações na aplicação.

## 5. Fluxos Alternativos

N/A (fluxo direto e simples).

## 6. Fluxos Secundários

### Formatar Dados para a Exibição

A aplicação organiza os dados de forma coesa antes de os apresentar ao utilizador.

## 7. Cenários Principais

### Administrador verifica ocupação na interface

O administrador visualiza o nível de ocupação diretamente na interface.

### Passageiro verifica ocupação na interface

O passageiro visualiza o nível de ocupação diretamente na interface.

## 8. Pós-Condições

Os dados do BRT são exibidos de forma clara e acessível ao utilizador.

## 9. Requisitos Especiais

A interface deve ser responsiva e acessível em vários dispositivos móveis.

## 6.6 Enviar por Meio de SMS as Informações do BRT

### 1. Descrição

Permite enviar informações do sistema BRT ao passageiro por via SMS.

### 2. Atores

Passageiro	Recebe as informações via SMS.
Aplicação	Gera e envia as informações solicitadas.

### 3. Pré-Condições

O passageiro deve ter um número de telemóvel registado na aplicação.

### 4. Fluxo Básico de Eventos

O passageiro solicita informações específicas (ex.: tempo de chegada).

A aplicação gera uma mensagem contendo os dados solicitados.

O sistema envia a mensagem para o número de telemóvel do passageiro.

### 5. Fluxos Alternativos

#### Erro no envio de SMS

A aplicação informa o erro e permite o reenvio da mensagem.

### 6. Fluxos Secundários

#### Gerar a Mensagem de Texto

A aplicação formata os dados solicitados em formato de texto.

### 7. Cenários Principais

#### Passageiro solicita tempo de chegada por via SMS

A aplicação envia o tempo estimado para o telemóvel do passageiro.

### 8. Pós-Condições

As informações são enviadas ao passageiro por SMS.

## 9. Requisitos Especiais

O sistema deve suportar a integração com serviços de SMS.

### 7.1 Selecionar Método de Pagamento

#### 1. Descrição

Este caso de uso descreve o processo pelo qual o passageiro escolhe o método de pagamento para realizar o pagamento.

#### 2. Ator

Passageiro	Responsável por selecionar o método de pagamento.
------------	---

#### 3. Pré-condições

O passageiro tem acesso ao sistema de pagamento.

O passageiro deve estar na página de verificação de pagamento.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando o passageiro acede à página de pagamento.

O passageiro visualiza os métodos de pagamento disponíveis (ex: MB Way, PayPal, etc.).

O passageiro escolhe o método de pagamento desejado.

O caso de uso termina quando o passageiro confirma a escolha do método de pagamento.

#### 5. Fluxos Alternativos

##### Método de pagamento não disponível

Se o método de pagamento escolhido não estiver disponível:

O passageiro é notificado de que o método não está disponível.

O passageiro escolhe outro método de pagamento.

#### 6. Fluxos Secundários

O sistema exibe os métodos de pagamento suportados e disponíveis para o passageiro.

#### 7. Cenários Principais

O passageiro escolhe e confirma o método de pagamento.

## 8. Pós-condições

O método de pagamento foi selecionado com sucesso.

## 9. Requisitos Especiais

O sistema deve ser intuitivo e rápido para que o passageiro consiga selecionar o método desejado sem dificuldades.

## 7.2 Efetuar Pagamento

### 1. Descrição

Este caso de uso descreve o processo em que o passageiro e o sistema de pagamento realizam a transação de pagamento.

### 2. Ator

Passageiro	O ator que insere os dados necessários para efetuar o pagamento.
Sistema de Pagamento	O sistema que processa os dados de pagamento fornecidos pelo passageiro.

### 3. Pré-condições

O passageiro selecionou o método de pagamento.

O passageiro tem os dados necessários para o pagamento (ex: número do cartão, dados de conta, etc.).

### 4. Fluxo Básico de Eventos

O caso de uso começa quando o passageiro insere os dados de pagamento na plataforma.

O sistema de pagamento recebe e valida os dados de pagamento.

O sistema de pagamento processa o pagamento.

O caso de uso termina quando o pagamento é processado com sucesso.

### 5. Fluxos Alternativos

#### Dados inválidos

Se os dados fornecidos pelo passageiro forem inválidos:

O sistema de pagamento solicita que o passageiro corrija os dados.

O passageiro insere os dados novamente e o pagamento é processado.

## 6. Fluxos Secundários

O sistema de pagamento valida a integridade dos dados fornecidos (ex: validade do cartão, saldo disponível).

## 7. Cenários Principais

O passageiro insere os dados e o sistema de pagamento processa o pagamento com sucesso

## 8. Pós-condições

O pagamento foi efetuado com sucesso.

## 9. Requisitos Especiais

O sistema de pagamento deve garantir a segurança dos dados e ser rápido no processamento.

## 7.3 Processar Pagamento

### 1. Descrição

Este caso de uso descreve o processo pelo qual o sistema de pagamento valida e processa o pagamento realizado pelo passageiro.

### 2. Ator

Sistema de Pagamento	Responsável por processar o pagamento.
----------------------	--

### 3. Pré-condições

O passageiro inseriu os dados de pagamento corretamente.

O sistema de pagamento está acessível e funcionando.

### 4. Fluxo Básico de Eventos

O caso de uso começa quando o sistema de pagamento recebe os dados de pagamento do passageiro.

O sistema de pagamento valida os dados inseridos.

O sistema de pagamento processa a transação, comunicando com o banco ou o serviço de pagamento.

O caso de uso termina quando o sistema de pagamento processa o pagamento.

### 5. Fluxos Alternativos

### **Falha na transação**

O sistema de pagamento notifica o passageiro sobre a falha.

O passageiro tenta outro método de pagamento ou corrige o erro.

### **6. Fluxos Secundários**

O sistema de pagamento comunica com o banco para verificar a disponibilidade de fundos.

### **7. Cenários Principais**

O sistema de pagamento processa o pagamento com sucesso.

### **8. Pós-condições**

O pagamento foi processado e a transação foi realizada.

### **9. Requisitos Especiais**

O sistema de pagamento deve garantir a integridade e segurança dos dados transacionados.

## **7.4 Validar Pagamento no Sistema Central**

### **1. Descrição**

Este caso de uso descreve o processo no qual o sistema de pagamento valida a transação no sistema central.

### **2. Ator**

Sistema de Pagamento	Responsável por validar a transação no sistema central.
Aplicação	Plataforma que verifica a confirmação do pagamento.

### **3. Pré-condições**

O sistema de pagamento processou o pagamento.

O sistema central está acessível para validação da transação.

### **4. Fluxo Básico de Eventos**

O caso de uso começa quando o sistema de pagamento envia os dados da transação para o sistema central.

O sistema central valida a transação com base nas informações recebidas.

O caso de uso termina quando o sistema central confirma a validação.

## 5. Fluxos Alternativos

### Falha na validação

Se a validação falhar:

O sistema de pagamento é notificado para tentar novamente.

O passageiro é informado sobre o erro e pode tentar um novo pagamento.

## 6. Fluxos Secundários

O sistema central verifica os dados da transação no banco de dados.

## 7. Cenários Principais

O sistema central valida a transação com sucesso.

## 8. Pós-condições

O pagamento foi validado no sistema central.

## 9. Requisitos Especiais

A validação no sistema central deve ser rápida e eficiente para evitar atrasos.

## 7.5 Emitir Comprovativo

### 1. Descrição

Este caso de uso descreve como a aplicação emite um comprovativo de pagamento para o passageiro.

### 2. Ator

Aplicação	Responsável por emitir o comprovativo.
-----------	--

### 3. Pré-condições

O pagamento foi validado com sucesso.

O passageiro deseja obter um comprovativo da transação.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando a aplicação recebe a confirmação de pagamento.  
A aplicação gera o comprovativo de pagamento.  
O passageiro recebe o comprovativo de pagamento por e-mail ou em formato digital.

#### 5. Fluxos Alternativos

##### **Falha na emissão**

Se ocorrer uma falha ao emitir o comprovativo:  
A aplicação tenta emitir o comprovativo novamente.

#### 6. Fluxos Secundários

A aplicação gera o comprovativo com os detalhes da transação (valor, data, número da transação).

#### 7. Cenários Principais

O passageiro recebe o comprovativo de pagamento.

#### 8. Pós-condições

O comprovativo foi gerado e entregue ao passageiro.

#### 9. Requisitos Especiais

A aplicação deve gerar comprovativos precisos e rápidos.

### 7.6 Enviar Notificação

#### 1. Descrição

Este caso de uso descreve como a aplicação envia uma notificação para o passageiro confirmando que o pagamento foi bem-sucedido.

#### 2. Ator

Aplicação	Responsável por enviar a notificação.
-----------	---------------------------------------

#### 3. Pré-condições

O pagamento foi validado e o comprovativo foi gerado.

A aplicação tem acesso ao sistema de envio de notificações (e-mail, SMS, etc.). O passageiro forneceu um método de contacto válido para o envio da notificação.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando a aplicação recebe a confirmação do pagamento. A aplicação gera a notificação com os detalhes do pagamento (valor, data, número da transação, etc.).

A aplicação envia uma notificação ao **passageiro** a confirmar o pagamento. O passageiro recebe a notificação através do método de envio preferido (e-mail, SMS, notificação push, etc.).

O caso de uso termina com a notificação enviada ao passageiro.

#### 5. Fluxos Alternativos

##### Falha na notificação

Se houver falha no envio da notificação:

A aplicação tenta novamente enviar a notificação.

Se o erro persistir, a aplicação regista a falha e o passageiro é notificado para tentar um outro método de contato.

#### 6. Fluxos Secundários

A aplicação envia uma mensagem (por e-mail, SMS, etc.) confirmando o pagamento.

#### 7. Cenários Principais

O passageiro recebe a notificação de pagamento bem-sucedido.

#### 8. Pós-condições

O passageiro foi notificado com sucesso.

#### 9. Requisitos Especiais

A aplicação deve garantir que a notificação seja entregue de forma eficiente.

### 8.1 Recolher Informações pertinentes

#### 1. Descrição

Este caso de uso descreve o processo no qual a aplicação recolhe informações pertinentes necessárias para gerar relatórios.

## 2. Ator

Aplicação	Responsável por recolher e organizar as informações pertinentes para os relatórios.
-----------	---

## 3. Pré-condições

A aplicação tem acesso aos dados necessários para gerar os relatórios.  
A aplicação está devidamente configurada para receber as informações.

## 4. Fluxo Básico de Eventos

O caso de uso começa quando a aplicação começa a recolher as informações necessárias.

A aplicação acede às fontes de dados pertinentes.

A aplicação organiza as informações recolhidas para posterior processamento.

O caso de uso termina quando as informações estão preparadas para gerar os relatórios.

## 5. Fluxos Alternativos

### Falha na recolha de dados

Se a aplicação não conseguir recolher as informações necessárias:

A aplicação tenta uma nova tentativa de recolha de dados.

Se a falha persistir, a aplicação alerta o administrador.

## 6. Fluxos Secundários

A aplicação acede a bases de dados ou sistemas externos para obter as informações necessárias.

## 7. Cenários Principais

A aplicação recolhe as informações necessárias para gerar os relatórios.

## 8. Pós-condições

As informações pertinentes foram recolhidas e estão prontas para gerar os relatórios.

## 9. Requisitos Especiais

A aplicação deve ser capaz de acessar rapidamente as fontes de dados e garantir a integridade das informações recolhidas.

## 8.2 Realizar Relatórios Operacionais

### 1. Descrição

Este caso de uso descreve o processo pelo qual o administrador gera relatórios operacionais utilizando as informações recolhidas pela aplicação.

### 2. Ator

Administrador	Responsável por gerar relatórios operacionais.
---------------	--

### 3. Pré-condições

O administrador tem acesso ao sistema de criação de relatórios.  
As informações necessárias foram recolhidas pela aplicação.

### 4. Fluxo Básico de Eventos

O caso de uso começa quando o administrador solicita a geração de relatórios operacionais.  
A aplicação utiliza as informações recolhidas para gerar o relatório.  
O administrador revê o relatório gerado.  
O caso de uso termina quando o relatório operacional é gerado com sucesso.

### 5. Fluxos Alternativos

#### Falha na geração do relatório

Se houver falha ao gerar o relatório:  
O administrador tenta gerar o relatório novamente.  
Se a falha persistir, o administrador solicita suporte técnico.

### 6. Fluxos Secundários

A aplicação utiliza os dados recolhidos para gerar o relatório de forma automática.

### 7. Cenários Principais

O administrador gera um relatório operacional com sucesso.

## 8. Pós-condições

O relatório operacional foi gerado com sucesso e está pronto para ser consultado.

## 9. Requisitos Especiais

O sistema de criação de relatórios deve ser capaz de gerar relatórios rápidos e precisos.

### 8.3 Realizar Relatórios Financeiros

#### 1. Descrição

Este caso de uso descreve o processo pelo qual o administrador gera relatórios financeiros utilizando as informações recolhidas pela aplicação.

#### 2. Ator

Administrador	Responsável por gerar relatórios financeiros
---------------	--

#### 3. Pré-condições

O administrador tem acesso ao sistema de geração de relatórios financeiros.  
As informações financeiras estão recolhidas e disponíveis na aplicação.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando o administrador solicita a realização de relatórios financeiros.

A aplicação utiliza as informações financeiras recolhidas para gerar o relatório.

O administrador revê o relatório financeiro gerado.

O caso de uso termina quando o relatório financeiro é gerado com sucesso.

#### 5. Fluxos Alternativos

##### **Falha na geração do relatório financeiro**

Se houver falha ao gerar o relatório:

O administrador tenta gerar o relatório novamente.

Se a falha persistir, o administrador solicita suporte técnico.

#### 6. Fluxos Secundários

A aplicação utiliza os dados financeiros recolhidos para gerar o relatório de forma automática.

#### 7. Cenários Principais

O administrador gera um relatório financeiro com sucesso.

#### 8. Pós-condições

O relatório financeiro foi gerado com sucesso e está pronto para ser consultado.

#### 9. Requisitos Especiais

O sistema de geração de relatórios financeiros deve ser capaz de criar relatórios rápidos e precisos, com todos os dados financeiros necessários.

### 8.4 Consultar os Relatórios

#### 1. Descrição

Este caso de uso descreve como o administrador consulta os relatórios gerados pela aplicação.

#### 2. Ator

Administrador	Responsável por consultar os relatórios.
Aplicação	Responsável por exibir os relatórios gerados.

#### 3. Pré-condições

O administrador tem acesso aos relatórios gerados pela aplicação.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando o administrador acessa o sistema para consultar os relatórios.

A aplicação exibe os relatórios gerados anteriormente.

O administrador seleciona o relatório desejado.

O caso de uso termina quando o administrador visualiza o relatório completo.

## 5. Fluxos Alternativos

### Relatório não encontrado

Se o relatório solicitado não for encontrado:

O administrador é notificado e pode tentar outro relatório.

## 6. Fluxos Secundários

A aplicação acessa a base de dados para exibir o relatório solicitado.

## 7. Cenários Principais

O administrador consulta o relatório de forma bem-sucedida.

## 8. Pós-condições

O administrador consultou com sucesso o relatório.

## 9. Requisitos Especiais

A aplicação deve permitir a consulta rápida e eficiente de relatórios.

## 8.5 Descarregar Relatórios

### 1. Descrição

Este caso de uso descreve como o administrador descarrega os relatórios gerados pela aplicação.

### 2. Ator

Administrador	Responsável por descarregar os relatórios.
Aplicação	Responsável por fornecer os relatórios para descarregamento.

### 3. Pré-condições

O administrador tem acesso ao relatório gerado.

O administrador está na interface para descarregar os relatórios.

### 4. Fluxo Básico de Eventos

O caso de uso começa quando o administrador seleciona a opção de descarregar um relatório.

A aplicação prepara o relatório para descarregamento.

O administrador inicia o descarregamento do relatório.

O caso de uso termina quando o relatório é descarregado com sucesso.

## 5. Fluxos Alternativos

### Falha no descarregamento

Se houver falha no descarregamento do relatório:

O administrador tenta descarregar o relatório novamente.

Se a falha persistir, o administrador solicita suporte técnico.

## 6. Fluxos Secundários

A aplicação prepara o relatório no formato adequado (ex: PDF, Excel) para ser descarregado.

## 7. Cenários Principais

O administrador consegue descarregar o relatório com sucesso.

## 8. Pós-condições

O relatório foi descarregado com sucesso.

## 9. Requisitos Especiais

A aplicação deve garantir que o processo de descarregamento seja rápido e sem falhas.

## 8.6 Agendar Futuros Relatórios

### 1. Descrição

Este caso de uso descreve como o administrador agenda relatórios futuros para serem gerados automaticamente pela aplicação.

### 2. Ator

Administrador	Responsável por agendar os relatórios.
---------------	--

### 3. Pré-condições

O administrador tem acesso ao sistema de agendamento de relatórios.

O administrador escolheu o tipo de relatório que deseja agendar.

#### 4. Fluxo Básico de Eventos

O caso de uso começa quando o administrador acede ao sistema de agendamento de relatórios.

O administrador escolhe o tipo de relatório a ser gerado e define o horário de agendamento.

A aplicação agenda o relatório para o horário definido.

O caso de uso termina quando o relatório é agendado com sucesso.

#### 5. Fluxos Alternativos

##### **Falha no agendamento**

Se houver falha ao agendar o relatório:

O administrador tenta agendar o relatório novamente.

Se a falha persistir, o administrador solicita suporte técnico.

#### 6. Fluxos Secundários

A aplicação verifica se os recursos necessários para gerar o relatório estão disponíveis no momento agendado.

#### 7. Cenários Principais

O administrador agenda um relatório com sucesso.

#### 8. Pós-condições

O relatório foi agendado para ser gerado no futuro.

#### 9. Requisitos Especiais

O sistema de agendamento deve ser flexível, permitindo agendar relatórios diários, semanais ou mensais.

## 1. Architecture Notebook

### 1. Objetivo

O Caderno de Arquitetura é um artefacto essencial para organizar e documentar os problemas relacionados com a arquitetura de um sistema, bem como as decisões tomadas para os resolver. Inclui elementos como, por exemplo, esquemas, padrões, código e outros recursos relevantes. As informações estão estruturadas em diferentes níveis, facilitando assim a compreensão das decisões arquitetónicas já tomadas e daquelas que ainda estão em curso.

Este artefacto desempenha um papel crucial ao proporcionar aos arquitetos uma ferramenta eficaz de colaboração com o restante da equipa, permitindo o desenvolvimento conjunto da arquitetura. Para os membros da equipa, o Caderno de Arquitetura oferece uma visão clara das razões que fundamentam as escolhas arquitetónicas, garantindo uma implementação sólida e bem sustentada.

Além disso, o Caderno de Arquitetura funciona como um guia sobre a divisão e a organização do sistema, permitindo que a equipa se adapte de forma eficiente às suas exigências. Também oferece uma visão inicial do sistema e das suas motivações técnicas, sendo valioso para quem, no futuro, precisar manter ou evoluir a arquitetura.

### 2. Objetivos e filosofia da arquitetura

A arquitetura do sistema foi projetada para priorizar a escalabilidade, a modularidade e o desempenho em tempo real, assegurando adaptação às exigências e uma evolução contínua. O design centra-se no utilizador, proporcionando uma experiência intuitiva, eficiente e acessível tanto para passageiros e motoristas quanto para administradores.

A escalabilidade é essencial para suportar o aumento da procura. A modularidade garante que o sistema possa ser atualizado e expandido de forma ágil, reduzindo custos e tempos de manutenção. O processamento em tempo real é igualmente crítico, oferecendo informações precisas sobre a localização dos autocarros e níveis de ocupação, o que aumenta a fiabilidade para os utilizadores. Adicionalmente, a segurança é um pilar central, com mecanismos robustos para proteção de dados sensíveis, como informações de pagamento e dados pessoais, em conformidade com as melhores práticas de cibersegurança.

O sistema deve assegurar uma experiência consistente e fluida em dispositivos móveis adaptando-se às diferentes necessidades dos utilizadores. Deve também integrar-se com sistemas de pagamento de terceiros, como MB Way e PayPal, de forma segura e eficiente. A gestão de dados em tempo real, provenientes dos sensores e GPS, é indispensável para garantir a relevância e a precisão das informações fornecidas.

Esta abordagem posiciona o sistema para responder às necessidades atuais e futuras, promovendo maior eficiência operacional e uma experiência de utilizador superior. A combinação de todas as funcionalidades, em tempo real, segurança avançada e usabilidade garante a sua relevância e competitividade no mercado.

### 3. Pressupostos e dependências

A implementação do sistema baseia-se em pressupostos sólidos e dependências essenciais, garantindo o seu funcionamento de forma eficiente e contínua. Parte-se do princípio de que os sensores e módulos GPS instalados nos autocarros fornecerão dados precisos e confiáveis para se saber a ocupação e localização em tempo real. Estes dados são fundamentais para disponibilizar informações exatas e atualizadas aos utilizadores. Outro pressuposto importante é o acesso a dispositivos inteligentes, como smartphones e computadores, tanto por parte dos passageiros e motoristas quanto dos administradores, o que garante uma interação eficiente com a plataforma.

A funcionalidade do sistema depende ainda de integrações críticas com serviços externos. Por exemplo, a compra de bilhetes requer uma integração segura e eficiente com sistemas de pagamento, como PayPal ou MB Way, garantindo uma experiência de pagamento simples e fiável para os utilizadores. Além disso, o envio de notificações, como alertas de horários ou atualizações de serviço, está vinculado a sistemas externos de mensagens SMS, que devem ser estáveis e bem integrados no sistema.

O desempenho e a disponibilidade do sistema também estão condicionados a uma infraestrutura confiável para hospedar os serviços de backend. Estas plataformas são essenciais para suportar o aumento do tráfego e oferecer uma base robusta para as operações do sistema. Por fim, o sucesso do projeto está diretamente ligado à proficiência da equipa de desenvolvimento em padrões arquitetónicos que asseguram um design modular e escalável, facilitando a manutenção e a expansão do sistema.

Com pressupostos claros e dependências bem definidas, o sistema estará preparado para atender às necessidades dos utilizadores, oferecendo uma experiência fluida e eficiente. A combinação de dados confiáveis, conectividade consistente, infraestrutura robusta e uma equipa qualificada permite a criação de uma solução tecnológica robusta, capaz de responder às exigências atuais e adaptar-se aos desafios do futuro.

### 4. Requisitos significativos do ponto de vista da arquitetura

Os requisitos significativos do ponto de vista da arquitetura do sistema definem os aspectos cruciais que influenciam o design e a implementação da solução. Estes requisitos baseiam-se nos diagramas de caso de uso e especificações funcionais, garantindo que o sistema atenda às necessidades dos utilizadores de forma eficiente e escalável.

### 5. Decisões, restrições e justificações

As decisões e restrições adotadas na arquitetura do sistema foram cuidadosamente planeadas para garantir escalabilidade, segurança e uma experiência eficiente para os utilizadores. Uma das principais decisões foi a adoção de uma arquitetura de

microsserviços, que modulariza funcionalidades como a gestão de bilhética e a localização em tempo real. Com esta abordagem, cada funcionalidade é implementada como um serviço independente, permitindo uma escalabilidade horizontal eficiente e simplificando a manutenção. Além disso, futuros módulos podem ser adicionados ou atualizados sem impactar os componentes já existentes, promovendo mais flexibilidade e robustez na solução.

Outra decisão importante foi o uso de APIs para a comunicação entre o frontend e o backend. Este modelo de interação foi escolhido por ser leve, escalável e agnóstico à plataforma, permitindo uma integração eficiente entre os diferentes componentes do sistema.

Entre as restrições técnicas estabelecidas, destaca-se o limite de 2 segundos para a conclusão de operações em tempo real, como atualizações de localização de autocarros. Este tempo de resposta foi definido para assegurar uma experiência fluida e confiável, atendendo às expectativas dos passageiros que dependem de informações atualizadas para efetuarem decisões. Outra restrição fundamental é o uso obrigatório de comunicação criptografada, como HTTPS para as APIs. Esta medida visa proteger os dados dos utilizadores, especialmente durante transações sensíveis, como pagamentos e autenticação.

## 6. Mecanismos de arquitetura

Os mecanismos arquiteturais do sistema foram concebidos para suportar funcionalidades críticas, garantindo maior desempenho, segurança e fiabilidade. Estes mecanismos são componentes essenciais para a operacionalização das principais funcionalidades do sistema.

### Mecanismo de Fluxo de Dados em Tempo Real

O mecanismo de fluxo de dados em tempo real tem como propósito processar e exibir informações atualizadas provenientes do GPS e dos sensores instalados nos autocarros. Este mecanismo é projetado para assegurar a baixa latência e a alta disponibilidade, características indispensáveis para garantir que os utilizadores recebem informações precisas e oportunas sobre a localização e ocupação dos veículos. A sua função principal é recolher dados diretamente dos autocarros, processá-los em tempo real e atualizá-los nas interfaces do utilizador. Para cumprir estes objetivos, o mecanismo deve integrar tecnologias de streaming de dados, armazenamento temporário eficiente e protocolos otimizados para comunicação em tempo real.

### Mecanismo de Processamento de Pagamento

O mecanismo de processamento de pagamento é responsável por gerir as transações relacionadas com os pagamentos, garantindo segurança, rapidez e confiabilidade. Este mecanismo integra-se com gateways de pagamento, como o MB Way e PayPal, para possibilitar transações fluidas e seguras. Assegura a proteção de dados sensíveis dos utilizadores durante todo o processo de pagamento, utilizando protocolos criptografados e conformidade com normas de segurança. Além disso, o mecanismo deve ser capaz de

lidar com grandes volumes de transações, mantendo um desempenho consistente e uma experiência positiva para o utilizador.

## 7. Principais abstrações

As abstrações principais do sistema representam os elementos essenciais para a sua funcionalidade. Estas entidades são a base para o design e implementação, permitindo uma organização lógica e eficiente das operações.

### **Passageiro**

O passageiro é uma entidade fundamental que interage diretamente com as funcionalidades do sistema. Esta abstração inclui atributos como identificação do utilizador, histórico de transações, etc. O passageiro utiliza o sistema principalmente para gestão de viagens e obter informações atualizadas sobre os serviços.

### **Administrador**

O administrador é o utilizador responsável pela gestão do sistema, com acesso a funcionalidades avançadas, como a configuração de autocarros, definição de horários e criação de relatórios operacionais. Esta entidade tem privilégios específicos para monitorizar o desempenho do sistema e tomar decisões baseadas nos dados recolhidos, assegurando o bom funcionamento do serviço.

### **Autocarro**

O autocarro é a entidade central do sistema, representando os veículos monitorizados e geridos. Possui atributos como a localização em tempo real, o nível de ocupação e os horários programados. Esta abstração está diretamente ligada aos dados fornecidos pelos sensores e aos serviços de GPS, sendo uma peça chave para a operação e visualização do sistema.

### **Bilhete**

O bilhete é uma abstração que representa uma tarifa adquirida pelos passageiros, associando-os a uma viagem específica. Contém informações como tipo de bilhete, validade e dados do autocarro correspondente. Esta entidade é crucial para a gestão da bilhética e integração com os mecanismos de pagamento.

### **Sensores**

Os sensores são um dispositivo físico que fornece dados em tempo real sobre a localização dos autocarros e o nível de ocupação. Esta abstração está ligada diretamente ao mecanismo de fluxo de dados em tempo real, recolhendo e enviando informações essenciais para que o sistema as possa processar e disponibilizar aos utilizadores.

## 8. Camadas ou estrutura arquitetónica

A arquitetura do sistema é organizada em camadas estruturadas e utiliza frameworks e padrões que asseguram escalabilidade, modularidade e eficiência. A camada de apresentação é responsável pelas interfaces dos diferentes tipos de utilizadores. Utilizando o padrão MVC (Model-View-Controller), esta camada separa a lógica de exibição da lógica de controlo e dos modelos de dados, facilitando a manutenção e evolução das interfaces.

A camada de lógica de negócios concentra a inteligência do sistema, processando solicitações dos utilizadores e executando regras de negócio. Esta camada é responsável por calcular os tempos estimados de chegada dos autocarros, validar bilhetes, gerir dados de ocupação e enviar notificações. Implementada como uma coleção de microsserviços independentes, esta abordagem promove modularidade, escalabilidade horizontal e maior resiliência, permitindo a atualização ou substituição de serviços individuais sem impacto significativo no sistema.

A camada de dados, por sua vez, é responsável pela gestão de informações persistentes, como registo de utilizadores, bilhetes, localização de autocarros e dados históricos. Utiliza bases de dados relacionais e/ou não relacionais, dependendo da natureza dos dados, garantindo alta disponibilidade e integridade. Além disso, suporta processos de recuperação e backup, assegurando que os dados estejam sempre protegidos e acessíveis.

O frontend segue o padrão MVC, permitindo uma separação clara entre a interface, a lógica de controlo e os dados, o que facilita na manutenção e implementação de novas funcionalidades. No backend, a arquitetura baseada em microsserviços permite que cada funcionalidade seja tratada de forma independente. Esta abordagem aumenta a tolerância a falhas, a escalabilidade e a flexibilidade para adotar diferentes tecnologias em serviços específicos.

Com esta estrutura, a arquitetura do sistema é capaz de atender às necessidades atuais e futuras, assegurando uma experiência fluida para os utilizadores, manutenção simplificada e robustez no backend, proporcionando um desempenho consistente e confiável.

## 9. Visão arquitectónica

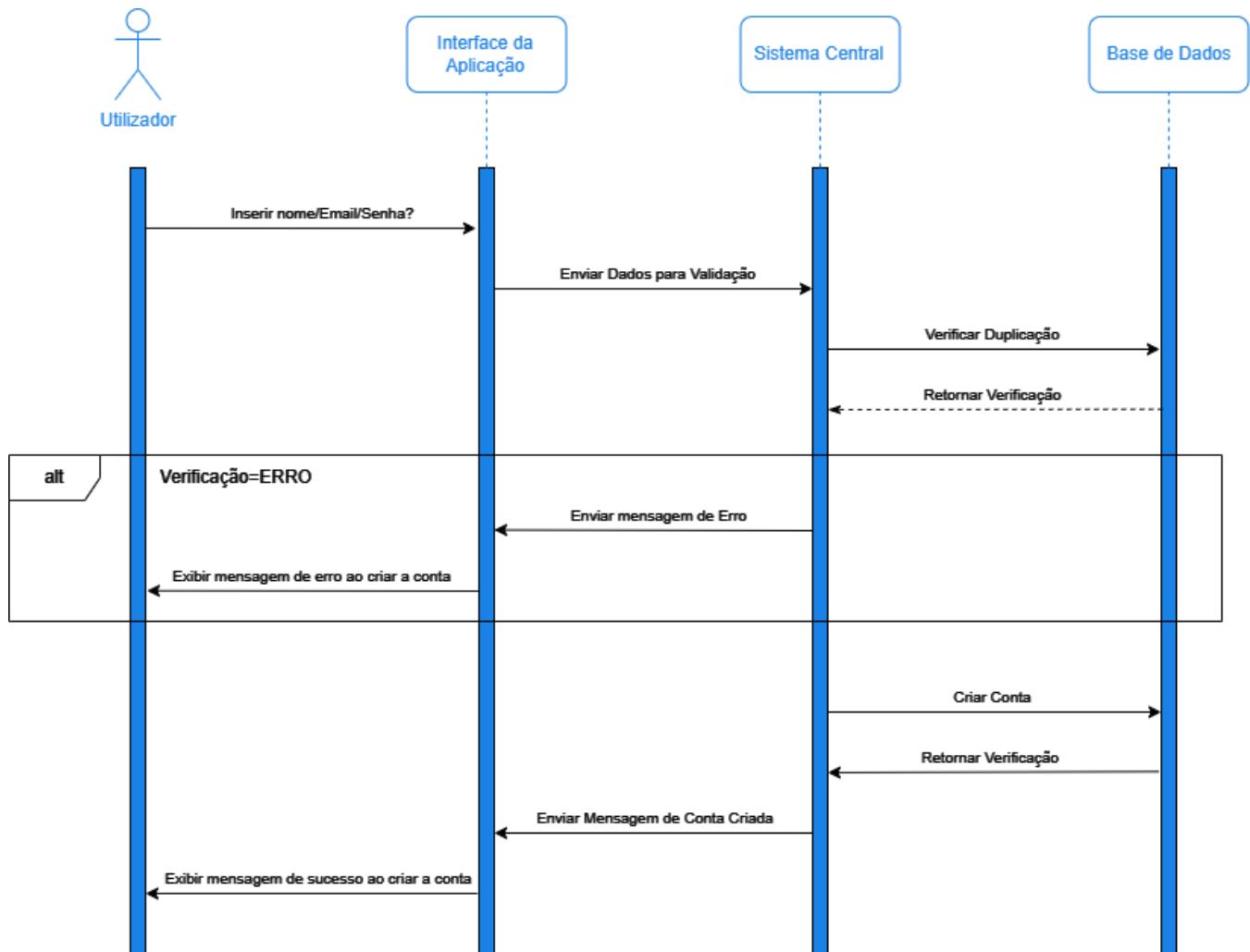


Diagrama 1- Criar Conta na Aplicação

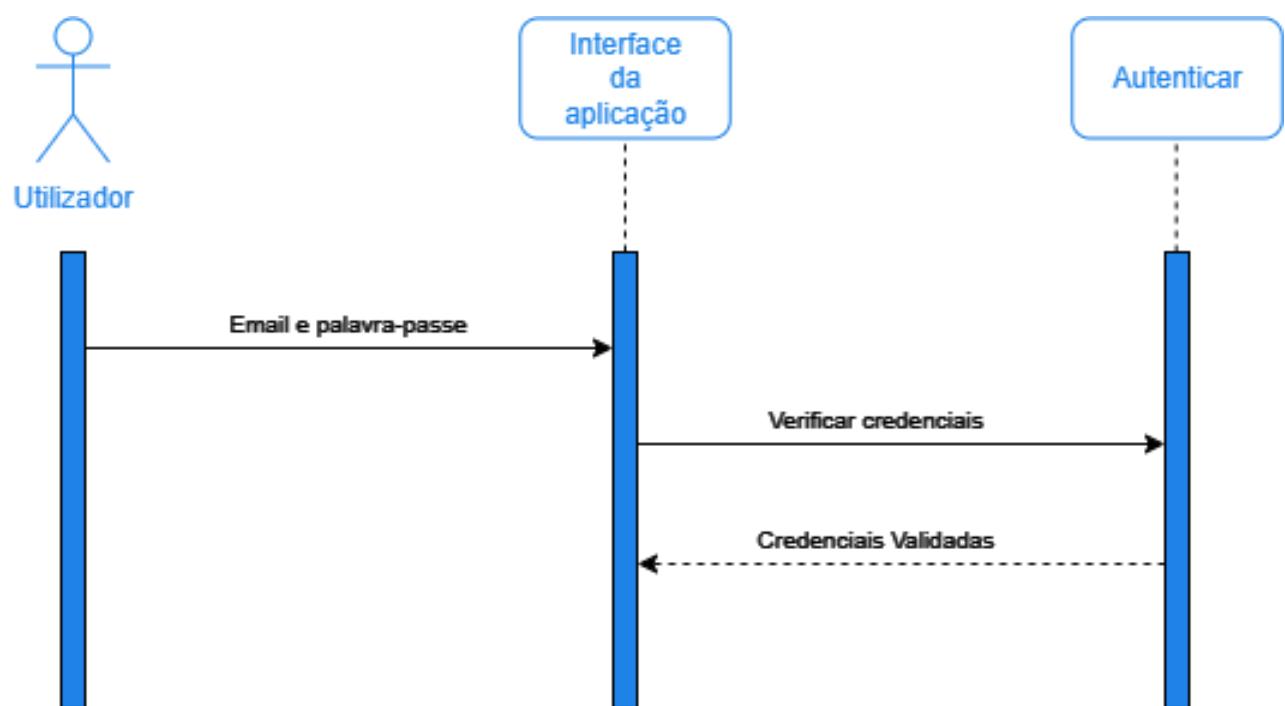


Diagrama 2 - Autenticar na Aplicação



Diagrama 3 - Editar Dados de Utilizador

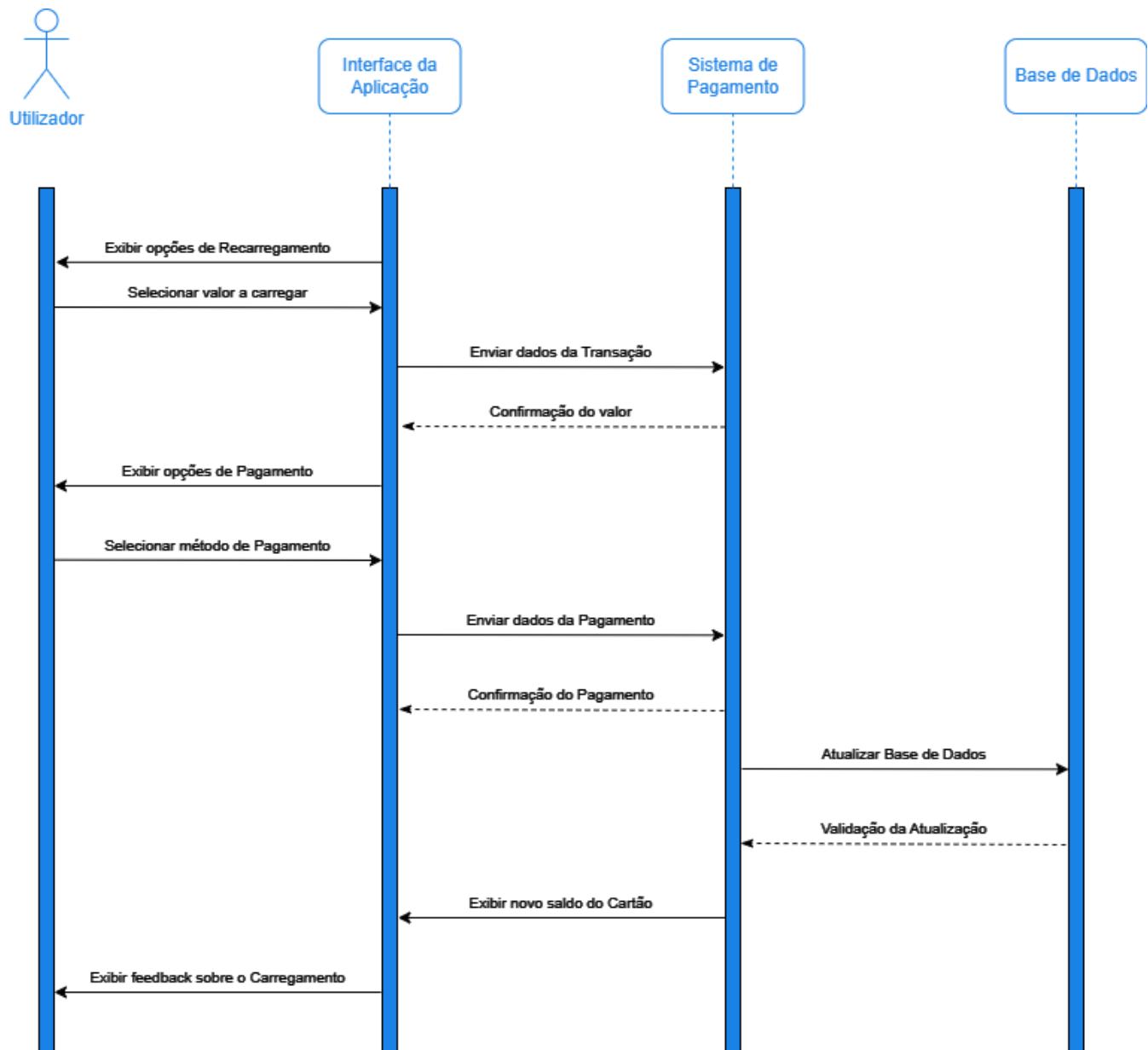


Diagrama 4 - Gerir os Créditos do Cartão

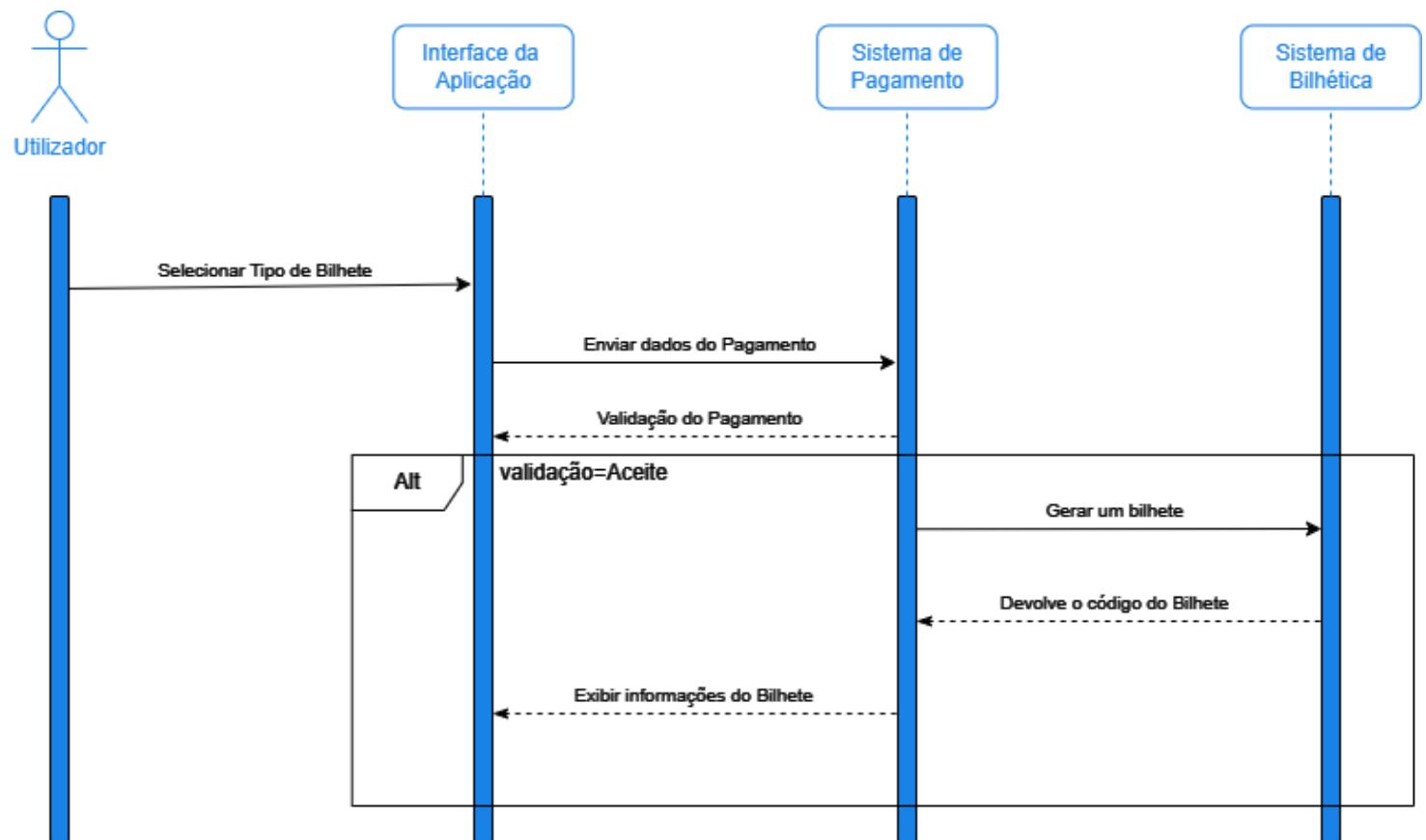


Diagrama 5 - Comprar um Bilhete

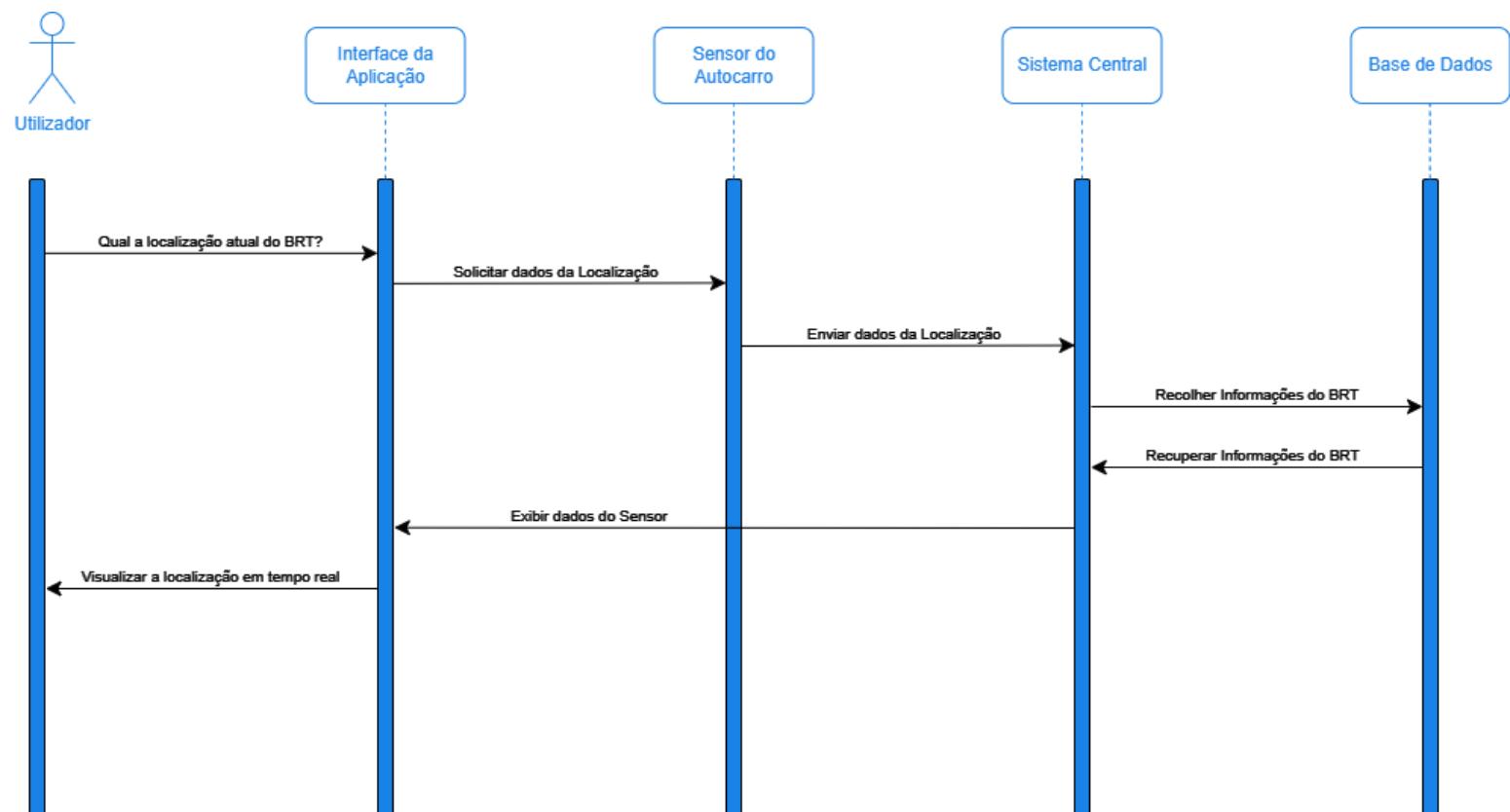


Diagrama 6 - Monitorizar o BRT

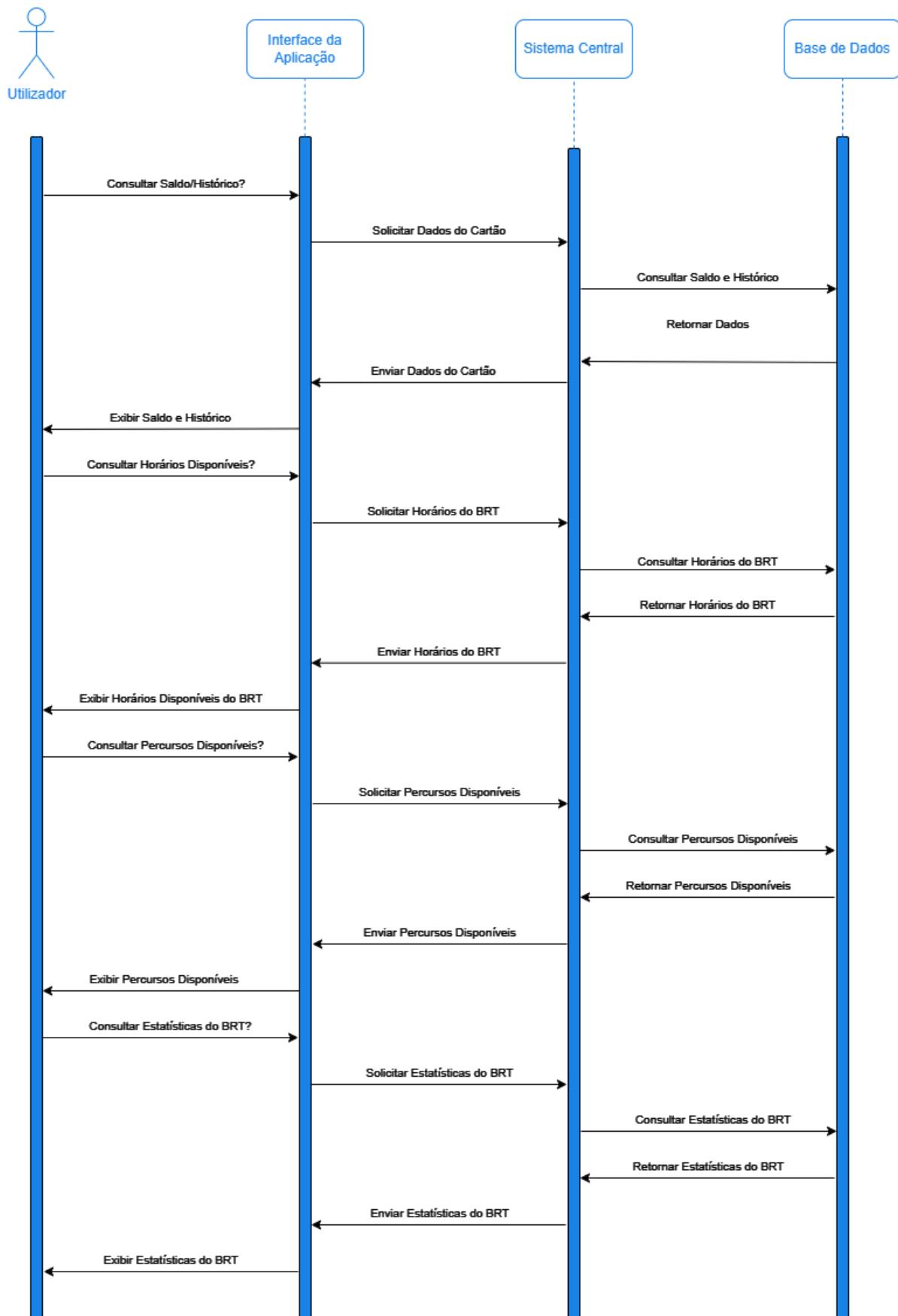


Diagrama 7 - Consultar a Aplicação

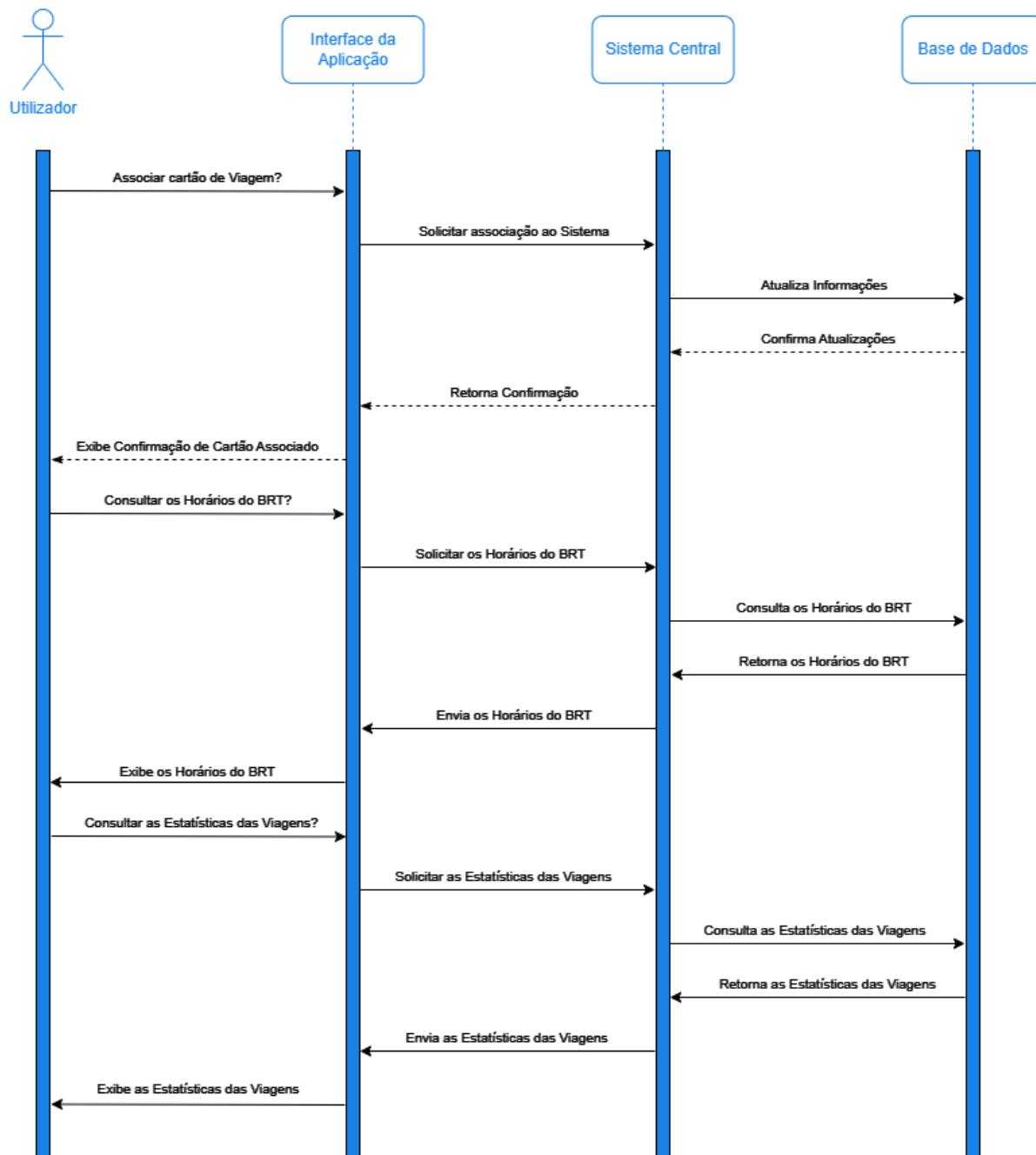


Diagrama 8 - Consultar Horários

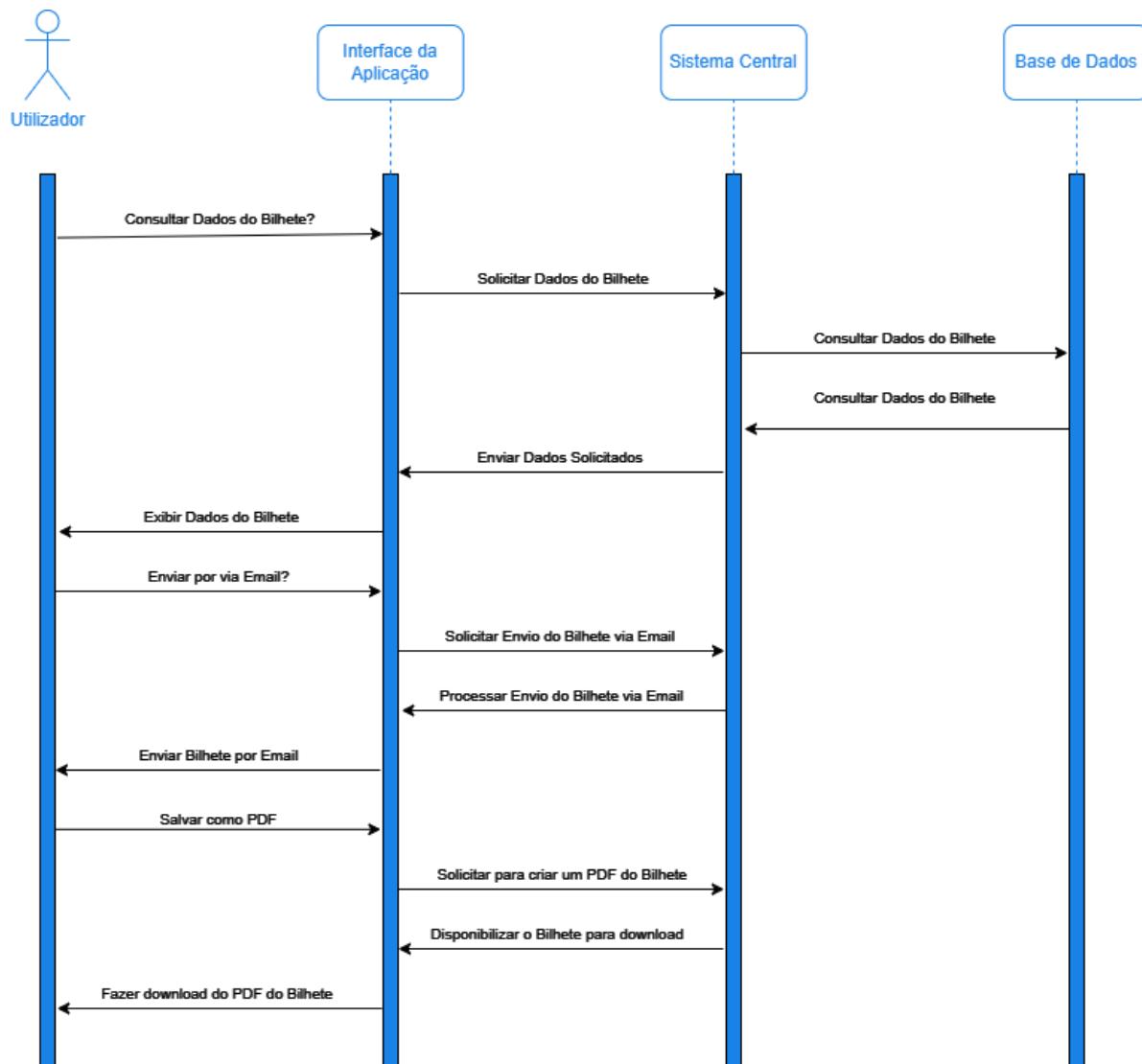


Diagrama 9 - Consultar Bilhete

## 2. Test Case

### 1. Criação de Conta

Description: Avalia se o utilizador consegue criar uma conta com os dados fornecidos (nome, email, palavra-passe e foto de perfil). O resultado esperado é que a conta seja criada com sucesso e o utilizador seja redirecionado para a página de login.

Pre-conditions: O utilizador deve estar na página de criação de conta, pronto para fornecer as informações necessárias. Não deve existir uma conta com o mesmo email no sistema.

Post-conditions: A conta é criada corretamente e o utilizador é informado sobre o sucesso do registo, podendo fazer login imediatamente. Caso haja algum erro (por exemplo, email já registado), o sistema deve exibir uma mensagem apropriada de erro.

Data required: Nome do utilizador, email válido, palavra-passe válida (mínimo de caracteres, requisitos de segurança), foto de perfil (opcional).

### 2. Login com Credenciais Válidas

Description: Este caso de teste verifica se o utilizador consegue autenticar-se corretamente no sistema ao fornecer um nome de utilizador ou email e a palavra-passe correspondente. O objetivo é garantir que o sistema aceite credenciais válidas e permita o acesso do utilizador. O resultado esperado é que, após a autenticação bem-sucedida, o utilizador seja redirecionado para a página inicial da aplicação.

Pre-conditions: O utilizador deve ter uma conta registada no sistema e estar na página de login. O nome de utilizador ou email e a palavra-passe fornecidos devem ser válidos.

Post-conditions: O sistema valida as credenciais, autentica o utilizador e redirecciona-o para a página inicial. Caso as credenciais sejam inválidas, o sistema exibe uma mensagem de erro.

Data required: Nome de utilizador ou email e palavra-passe válidos.

### 3. Login com Credenciais Inválidas

Description: Verifica se o sistema rejeita tentativas de login com credenciais inválidas, fornecendo feedback claro ao utilizador sobre o motivo da rejeição. Este teste avalia se o sistema exibe mensagens apropriadas para credenciais incorretas e impede o acesso não autorizado.

Pre-conditions: O sistema deve estar funcional e acessível, com a funcionalidade de login configurada e ativa.

Post-conditions: O sistema não deve permitir o acesso ao utilizador, garantindo que uma mensagem clara, como "Nome de utilizador ou senha incorreta", seja apresentada, enquanto mantém a conta protegida contra acessos não autorizados.

Data required: Credenciais inválidas, como um nome de utilizador ou e-mail inexistente, ou uma palavra-passe incorreta associada a uma conta válida.

### 4. Verificar Compra de Bilhete "Viagem Única"

Description: Verificar se o sistema permite que o utilizador efetue a compra de um bilhete de "Viagem Única", garantindo que o bilhete é registado corretamente na conta do utilizador e que o pagamento é processado sem erros. O teste também avalia se o utilizador recebe um recibo da transação, confirmando a aquisição.

Pre-conditions: O utilizador deve estar autenticado no sistema, ter um método de pagamento válido configurado e saldo disponível, se aplicável. Além disso, a opção de bilhete "Viagem Única" deve estar ativa e acessível no sistema.

Post-conditions: Após a compra, o bilhete deve ser disponibilizado no histórico de bilhetes do utilizador e estar pronto para utilização. O pagamento deve ser registado corretamente, e um recibo deve ser gerado e armazenado no sistema.

Data required: O teste requer o tipo de bilhete selecionado ("Viagem Única") e os detalhes do método de pagamento utilizado pelo utilizador.

## 5. Verificar Compra de Bilhete "Passe Diário"

Description: Confirmar que o sistema permite a compra de um passe diário, processando o pagamento corretamente e ativando o passe na conta do utilizador para utilização imediata. Este teste verifica o fluxo completo, desde a seleção do bilhete até à sua ativação no sistema.

Pre-conditions: O utilizador deve estar autenticado no sistema, possuir um método de pagamento válido configurado ou saldo suficiente para a transação. O sistema deve ter a opção de "Passe Diário" ativa e disponível para compra.

Post-conditions: O passe diário deve ser ativado com sucesso e visível na conta do utilizador. A transação deve ser registada corretamente no sistema, e um recibo deve ser gerado como comprovativo da compra.

Data required: O teste utiliza o tipo de bilhete selecionado ("Passe Diário") e os dados do método de pagamento fornecido pelo utilizador.

## 6. Verificar Efetuação de Pagamento com Método de Pagamento Válido

Description: Garantir que o sistema processa corretamente pagamentos realizados com métodos válidos, assegurando que as transações são concluídas sem erros e que os registo associados são atualizados no sistema. Este teste também verifica se o utilizador recebe uma confirmação adequada da transação.

Pre-conditions: O utilizador deve ter um método de pagamento válido previamente configurado, como MB Way ou PayPal, e o sistema deve estar integrado com a plataforma de pagamento correspondente e funcional para processar a transação.

Post-conditions: O pagamento deve ser concluído com sucesso, a transação registada no histórico do utilizador, e o sistema deve gerar um recibo da compra. O saldo ou os dados do método de pagamento do utilizador devem refletir corretamente o valor da transação.

Data required: O teste requer o método de pagamento selecionado e o valor da transação correspondente ao serviço ou bilhete adquirido.

## 7. Verificar Falha no Pagamento

Description: Verificar se o sistema lida adequadamente com falhas no pagamento, como saldo insuficiente ou métodos inválidos, assegurando que a transação não é concluída e que o utilizador recebe uma notificação clara sobre o motivo da rejeição.

Pre-conditions: O utilizador deve tentar realizar um pagamento utilizando um método inválido ou com saldo insuficiente. O sistema deve estar funcional e conectado às plataformas de pagamento, apto a verificar o estado da transação.

Post-conditions: O pagamento deve ser rejeitado, a transação marcada como falhada no sistema, e o utilizador deve visualizar uma mensagem clara de erro, como "Pagamento recusado devido a saldo insuficiente". Nenhuma cobrança deve ser aplicada.

Data required: Para o teste, são necessários os detalhes do método de pagamento inválido ou com saldo insuficiente e o valor da transação a ser realizada.

## 8. Verificar Consulta de Horários

Description: Garantir que o sistema permite ao utilizador consultar os horários disponíveis dos autocarros em tempo real, assegurando que as informações exibidas são precisas e atualizadas. Este teste avalia a fiabilidade dos dados apresentados e a usabilidade da funcionalidade de consulta.

Pre-conditions: O sistema deve estar funcional e ligado à base de dados de horários, com informações atualizadas sobre as rotas e os autocarros..

Post-conditions: Os horários dos autocarros devem ser apresentados corretamente, incluindo quaisquer alterações ou atrasos em tempo real. O utilizador deve conseguir selecionar uma linha ou rota específica e visualizar os detalhes de forma clara.

Data required: O teste utiliza a identificação de uma linha de autocarro ou rota específica para filtrar e exibir os horários correspondentes.

## 9. Verificar Localização em Tempo Real do Autocarro

Description: Garantir que o sistema permite ao utilizador visualizar a localização de um autocarro em tempo real, apresentando informações precisas e atualizadas sobre o trajeto. Este teste avalia a integração com os dispositivos de GPS instalados nos autocarros e a interface de visualização.

Pre-conditions: O sistema deve estar integrado com os dispositivos de GPS dos autocarros e a funcionalidade de localização em tempo real deve estar operacional. O utilizador deve ter acesso à aplicação e selecionar um autocarro ou rota específica.

Post-conditions: A localização do autocarro deve ser exibida de forma precisa num mapa interativo, com atualizações frequentes. O sistema deve notificar o utilizador em caso de falha na obtenção de dados ou se o autocarro não estiver em operação.

Data required: O teste requer o número do autocarro ou a rota selecionada pelo utilizador para obter os dados de localização associados.

## 10. Verificar Consulta de Bilhete

Description: Garantir que o utilizador pode consultar os bilhetes adquiridos e ativos na aplicação, incluindo informações detalhadas como validade, tipo e data de utilização. Este teste avalia a precisão dos dados apresentados e a funcionalidade de gestão de bilhetes na aplicação.

Pre-conditions: O utilizador deve estar autenticado e ter bilhetes previamente adquiridos associados à sua conta. O sistema deve estar funcional e com acesso ao histórico de bilhetes do utilizador.

Post-conditions: O sistema deve exibir corretamente uma lista de bilhetes comprados pelo utilizador, incluindo detalhes como estado (ativo, expirado, usado), validade e outras informações relevantes. O utilizador deve poder identificar facilmente o bilhete desejado para uso ou verificação.

Data required: O teste requer o acesso aos dados do histórico de bilhetes do utilizador, armazenados na sua conta, para exibir as informações necessárias.

## 11. Verificar Gestão de Perfil do Utilizador

Description: Garantir que o sistema permite ao utilizador gerir as informações do seu perfil, como alterar dados pessoais, métodos de pagamento e preferências de utilização. Este teste avalia se as alterações feitas são aplicadas corretamente e refletidas na conta do utilizador.

Pre-conditions: O utilizador deve estar autenticado no sistema, e a funcionalidade de edição de perfil deve estar ativa e operacional.

Post-conditions: As alterações realizadas pelo utilizador devem ser guardadas e refletidas na sua conta de forma imediata. O sistema deve fazer uma confirmação de sucesso após a atualização dos dados.

Data required: O teste utiliza os novos dados fornecidos pelo utilizador, como nome, endereço de e-mail, preferências de notificação ou alterações nos métodos de pagamento associados.

## 10. Proof of Concept

A *Proof of Concept* tem como objetivo principal verificar a viabilidade técnica de uma ideia de software, avaliando se pode ser construída de forma eficaz e sustentável, considerando custos e recursos disponíveis. Focada em pesquisas e testes iniciais, procura identificar soluções ideais e tecnologias adequadas para atingir os resultados desejados, economizando tempo e recursos durante o desenvolvimento.

Além disso, auxilia na validação da ideia junto aos stakeholders, proporcionando uma visão clara e prática do conceito proposto e reforçando sua viabilidade. Neste artefato, apresentamos os modelos visuais e funcionais do sistema que planeamos, ilustrando o conceito da aplicação a ser desenvolvida.



a) Logotipo da Mobil-PMS

## 1. Criar Conta

Esta tela é destinada aos novos utilizadores que desejem autenticar-se no sistema.

- **Inserir dados do utilizador:**
  - Campo para o **email** do utilizador.
  - Campo para o **nome** do utilizador.
  - Campo para a **palavra-passe** que será associada à conta.
- **Fotografia:**
  - Possibilidade de adicionar uma **fotografia de perfil**.
- **Opções de interação:**
  - O Botão para **Login**, permite que o utilizador aceda à sua conta existente, reencaminhando para a página do mesmo.
  - O Botão para **Criar Conta**, permite que novos utilizadores se registem no sistema.

### < Criar Conta

**Fotografia Utilizador**

Email

Nome

Palavra-Passe

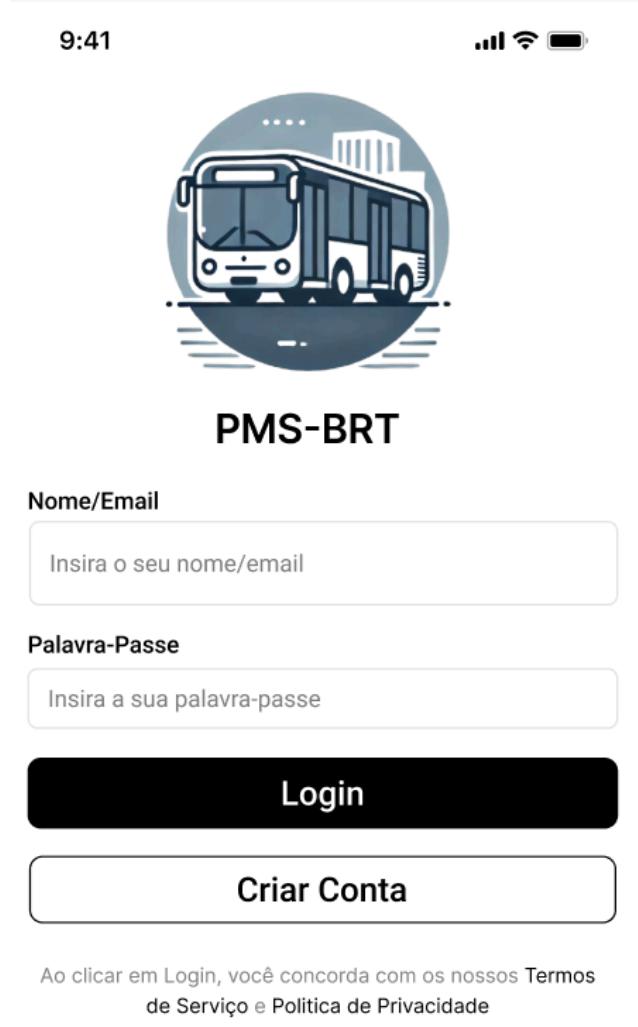
## 2. Autenticação

Este é o ponto de entrada para todos os passageiros, seja para login ou registo no sistema.

1. **Autenticação do utilizador:**
  - Campo para **inserir nome ou email**.
  - Campo para **inserir palavra-passe**.

## 2. Opções de interação:

- Botão **Login** para autenticar o utilizador com os dados fornecidos.
- Botão **Criar Conta**, que direciona para a página respetiva.



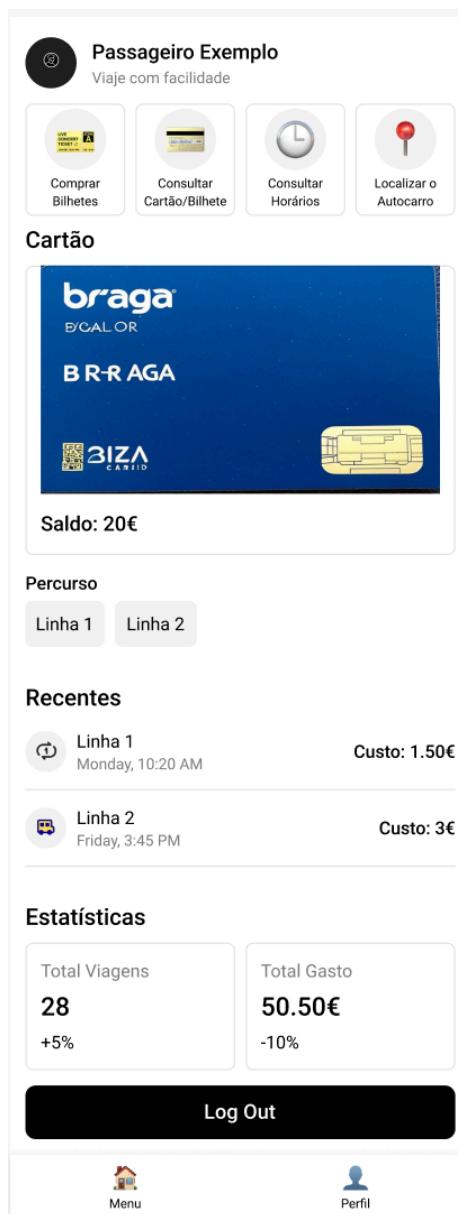
## 3. Página Inicial

Esta tela serve como um **dashboard completo**, fornecendo acesso às principais funcionalidades da aplicação, ao mesmo tempo que apresenta informações essenciais sobre saldo, viagens realizadas e estatísticas de uso. É o ponto central para a gestão da experiência do passageiro no sistema.

### 1. Ações Rápidas:

- **Comprar Bilhetes**: Opção para adquirir bilhetes que reencaminha para a respetiva página.
- **Consultar Cartão/Bilhete**: Permite verificar detalhes do cartão ou bilhete associado ao passageiro, reencaminhando para a respetiva página.
- **Consultar Horários**: Exibe os horários disponíveis para as linhas de autocarro e reencaminha para a página dos horários.

- **Localizar o Autocarro:** Acesso à localização em tempo real dos autocarros, reencaminha para a página específica.
- 2. **Cartão do Passageiro:**
  - Visualização do cartão associado ao passageiro.
  - **Saldo Atual:** Mostra o valor disponível no cartão para pagamento.
- 3. **Percorso:**
  - Botões para alternar entre **Linha 1** e **Linha 2**, personalizando a visualização dos detalhes associados às viagens realizadas nessas linhas.
- 4. **Recente:**
  - Exibição de viagens recentes, com detalhes como:
    - Nome da linha.
    - Data e hora da viagem.
    - Custo associado.
- 5. **Estatísticas:**
  - **Total de Viagens:** Número total de viagens realizadas, com variação percentual em relação a períodos anteriores.
  - **Total Gasto:** Gasto acumulado em viagens, com variação percentual indicando aumento ou redução nos custos.
- 6. **Opções de Interação:**
  - Botão para sair da aplicação de forma segura.
  - **Menu:** Acesso à página principal ou funcionalidades relacionadas à navegação geral.
  - **Perfil:** Acesso às configurações e informações pessoais do passageiro.



## 4. Compra de bilhetes

Esta tela facilita o processo de aquisição de bilhetes, permitindo que os usuários escolham o tipo de bilhete ou cartão desejado, ajustem a quantidade e associem o bilhete à linha desejada. É uma interface simples e eficiente, com detalhes claros sobre a validade e os custos dos bilhetes, oferecendo flexibilidade e praticidade ao usuário.

### 1. Tipos de Bilhetes:

- **Comprar Viagem:** Opção para adquirir bilhetes de viagem única.
- **Cartão BRT:** Opção para comprar cartões recarregáveis ou múltiplas viagens.

**2. Seleção de Número de Viagens:**

- Campo para inserir o número de viagens desejadas, com limite máximo de 5 viagens por compra.

**3. Escolha da Linha:**

- Botões de seleção para escolher entre **Linha 1** e **Linha 2**, permitindo que o bilhete seja associado à linha correta.

**4. Botão "Comprar":**

- Inicia o processo de compra do bilhete selecionado.

**5. Bilhetes Disponíveis:**

- **Bilhete de Viagem Única:**
  - Informações sobre a validade do bilhete.
  - Preço do bilhete
- **Cartão de 10 Viagens:**
  - Validade do cartão até uma data específica.
  - Preço do cartão

< Comprar Bilhetes

Comprar Viagem Cartão BRT

Número de viagem

e.x. 2

Máximo de 5 viagens

Linha

Linha 1 Linha 2

Comprar

Bilhetes disponíveis

Expira em 2 horas	Válido até 30/12/2024
1 viagem - linha 1	Cartão com 10 viagens
Preço: 2.50€	Preço: 20.00€

## 5. Pagamento

Esta tela facilita o processo de pagamento, oferecendo flexibilidade nas formas de pagamento e a possibilidade de customizar a compra com a adição de um NIF ou código

promocional. Permite que os utilizadores visualizem os custos, escolham o bilhete apropriado e utilizem o método de pagamento mais conveniente.

**1. Fatura:**

- Campo para inserir o **NIF (Número de Identificação Fiscal)**.

**2. Tipo de Bilhete:**

- **Viagem Única:**

- Custo
- Ideal para uma única utilização no transporte.

- **Passe Diário:**

- Custo
- Permite viagens ilimitadas durante um dia.

**3. Método de Pagamento:**

- Opções de métodos de pagamento disponíveis:

- **Cartão**.
- **Mbway**.
- **PayPal**.

**4. Resumo do Total:**

- Mostra o subtotal calculado com base no tipo de bilhete selecionado.

**5. Código Promocional:**

- Campo para inserção de um código promocional que pode oferecer descontos ou vantagens durante o pagamento.

**6. Efetuar Pagamento:**

- Botão que finaliza o processo de pagamento após as escolhas serem realizadas.

<  Pagamento

Fatura

 Inserir NIF

**Tipo de Bilhete**

 Viagem única 2.50€  Passe diário 7.00€
--

**Método de Pagamento**

Cartão   Mbway   PayPal

Escolha um método de pagamento

**Total**

SubTotal  
**8.50€**

**Código Promocional**

**Efetuar Pagamento**

## 6. Histórico de Viagens

O histórico de viagens permite que os passageiros tenham um controlo claro sobre as suas atividades recentes no sistema, com fácil acesso a informações detalhadas sobre as viagens, estatísticas e ações adicionais como comprar bilhetes ou consultar horários. É uma interface prática para ajudar os utilizadores a gerir e planejar as suas próximas viagens com eficiência.

### 1. Informações do Utilizador:

- Exibição do **nome** do passageiro e seu papel no sistema.

### 2. Viagens Recentes:

- Lista de viagens realizadas recentemente:
  - Exibe a linha, percurso e a data da viagem.
- Detalhamento fácil para que o utilizador acompanhe as suas viagens anteriores.

### 3. Ações Relacionadas:

- **Associar Cartão de Viagem:**
  - Permite vincular um cartão de transporte físico ou virtual à conta do passageiro.
- **Consultar Horários:**
  - Acesso rápido para visualizar horários.
- **Comprar Bilhetes:**
  - Direciona para a tela de compra de bilhetes.

### 4. Estatísticas de Viagem:

- **Resumo das Últimas Viagens:**
  - Total de viagens realizadas.
  - Distância total percorrida.
- Botão **Mais** para expandir e visualizar estatísticas adicionais, caso necessário.

The screenshot shows the 'Histórico de Viagens' (Travel History) screen. At the top, there's a back arrow and the title 'Histórico de Viagens'. Below that is a section for 'Nome' (Name) labeled 'Passageiro' with a placeholder icon. The next section, 'Viagens recentes', lists two trips: 'Linha 1 De Centro até Polo Universitário' (Line 1 From Center to University Pole) on 10/20/2021 and 'Linha 2 De ZonaA para ZonaB' (Line 2 From ZoneA to ZoneB) on 10/18/2021. Below these are three buttons: 'Associar Cartão de Viagem' (Associate Travel Card), 'Consultar Horários' (Check Schedules), and a large black 'Comprar Bilhetes' (Buy Tickets) button. At the bottom, there's a section for 'Estatísticas de Viagem' (Travel Statistics) with a 'Mais >' link. It shows a summary: 'Total de Viagens 5' and 'Km viajados 25 km'. At the very bottom are navigation links for 'Menu' and 'Perfil'.

## 7. Consulta de Horários

A consulta de horários oferece aos passageiros a capacidade de visualizar os horários de diferentes linhas, os utilizadores podem tomar decisões informadas para otimizar o seu tempo e selecionar a rota mais conveniente.

### 1. Seleção de Linha:

- Escolher entre as diferentes linhas disponíveis para visualizar os horários correspondentes.

### 2. Exibição dos Próximos Autocarros:

- Para cada linha e sentido, são apresentados os horários dos próximos autocarros.

Linha	Próximo autocarro	Horário
Linha 1	Rota A	5 minutos
	Rota A	20 minutos
	Rota B	10 minutos
Linha 2	Rota X	8 minutos
	Rota X	15 minutos
	Rota Y	12 minutos

## 8. Localização dos Autocarros em Tempo Real

A funcionalidade de localização em tempo real é essencial para que os passageiros acompanhem a posição do autocarro. Com informações detalhadas sobre a linha, rota, horário de chegada e localização atual, os utilizadores têm um maior controlo sobre o tempo de espera.

1. **Mapa de Localização:**

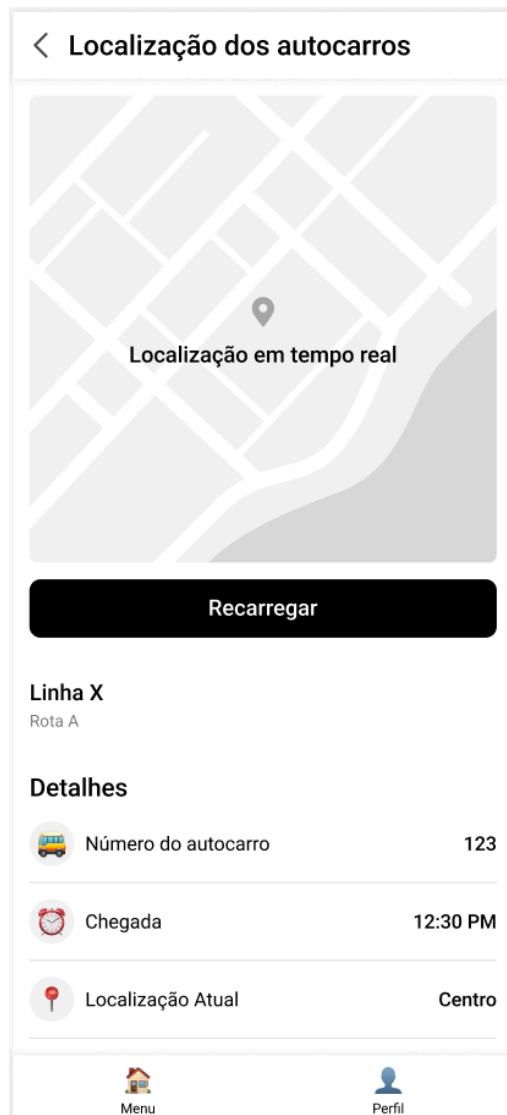
- Exibe um mapa com a **localização em tempo real** associada à linha selecionada.

2. **Botão de Atualização:**

- O botão "**Recarregar**" permite atualizar as informações de localização para obter dados mais recentes.

3. **Detalhes do Autocarro:**

- **Linha e Rota.**
- **Número do Autocarro.**
- **Horário de Chegada.**
- **Localização Atual.**



## 9. Consultar Bilhete

Esta funcionalidade permite aos utilizadores aceder rapidamente ao QR Code para validação do bilhete, verificarem informações essenciais como validade e local associado, e armazenarem o bilhete em diferentes formatos.

**1. Visualização do QR Code:**

- Apresenta o QR Code do bilhete, que pode ser utilizado para validação no sistema de transporte.

**2. Informações do Bilhete:**

- **Validade:** Indica a data de validade do bilhete.
- **Número do Bilhete:** Mostra o identificador único do bilhete.

**3. Opções de Compartilhamento e Salvamento:**

- **Enviar por Email:** Permite enviar uma cópia do bilhete para um endereço de email.
- **Salvar como PDF:** Oferece a opção de descarregar uma versão digital do bilhete em formato PDF.
- **Descarregar Bilhete:** Realiza o download direto do bilhete para armazenamento local.

**4. Detalhes do Bilhete:**

- **Data:** Mostra a data de compra ou emissão do bilhete.
- **Localização:** Indica o local associado ao bilhete, como ponto de embarque ou destino.
- **Quantidade:** Especifica o valor pago pelo bilhete.

< Consultar Bilhete

Informações do Bilhete

Validade: 01/01/2023



Numero do Bilhete: 123456

[Enviar por Email](#)

[Salvar como PDF](#)

[Descarregar bilhete](#)

Detalhes do Bilhete

	Data	12/12/2022
	Localização	Estação Central
	Quantia	25.00€

## 10. Perfil do passageiro

A funcionalidade do Perfil permite que o passageiro visualize as suas informações pessoais e histórico de viagens, além de gerir o seu cartão BRT e editar dados da conta. Esta secção promove a personalização e facilita o controlo de informações associadas ao uso do sistema de transporte.

### 1. Informações do Usuário:

- Mostra o nome do passageiro e o ano de registo no sistema.

### 2. Histórico de Viagens Recentes:

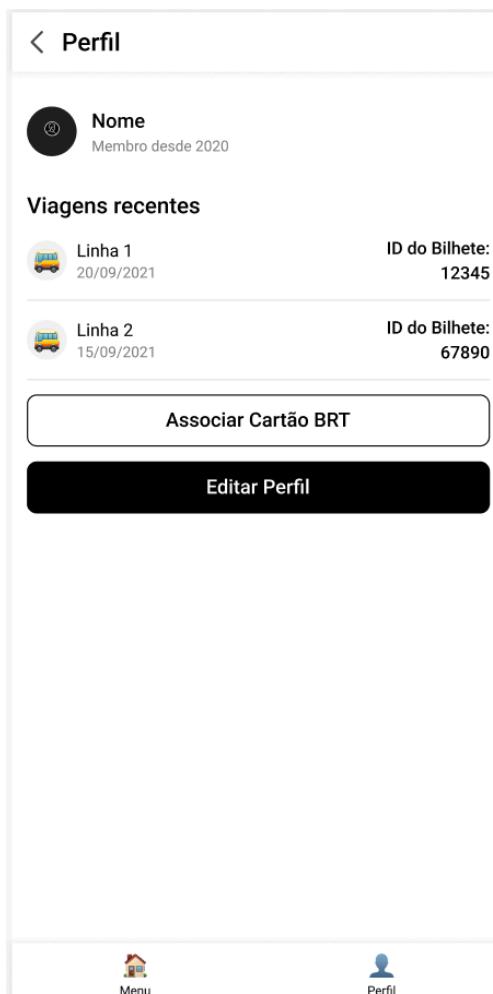
- Exibe as últimas viagens realizadas, organizadas por linha.
- Inclui detalhes como:
  - i. Identificação da linha utilizada.
  - ii. A data em que a viagem foi realizada.
  - iii. Número único do bilhete associado a cada viagem.

### 3. Gestão de Cartão BRT:

- **Botão Associar Cartão BRT:** Permite vincular um cartão BRT à conta do utilizador para facilitar a compra e o uso de bilhetes.

### 4. Edição de Perfil:

- **Botão Editar Perfil:** Oferece a funcionalidade de atualizar informações pessoais, como nome, email ou foto de perfil.



## 11. Editar Perfil

Esta página garante que o utilizador possa manter os seus dados atualizados de forma intuitiva. É uma funcionalidade essencial para melhorar a experiência do utilizador e a segurança da conta.

### 1. Alteração de Nome:

- Campo para inserir um novo nome para o perfil do utilizador.

### 2. Atualização de Email:

- Campo para adicionar ou modificar o endereço de email registado.

### 3. Alteração de Palavra-passe:

- Campo para definir uma nova palavra-passe para a conta.

### 4. Gerenciamento da Foto de Perfil:

- Espaço reservado para a Fotografia.

### 5. Ações de Confirmação ou Cancelamento:

- **Cancelar:** Descartar quaisquer alterações feitas e voltar ao menu anterior.
- **Guardar Alterações:** Salva as alterações feitas no perfil.

< Editar Perfil

Fotografia Utilizador  
User Profile

Nome  
Insira o novo nome  
Necessário

Email  
Insira o novo email  
Necessário

Palavra-passe  
Insira a nova palavra-passe

Cancelar Guardar alterações

Menu Perfil...