

Instituto Politécnico do Cávado e do Ave

Curso Técnico Superior Profissional -
Desenvolvimento Web e Multimédia

Algoritmos e Estrutura de Dados
Prof. Eng. José Salgado

Autores:

(Nome1 + nº de aluno)
(Nome2 + nº de aluno)
Telmo Regalado N°30539

Índice

1. Introdução	
1.1 Tema , descrição e objetivo do projeto.....	3
2. Desenvolvimento	
2.1 Abordagem e definição do problema.....	5
2.2 Plano de ação.....	5
2.3 Criação das estruturas do programa.....	6
2.4 Funções - Gestão de Pórticos.....	7
2.5 Funções - Gestão de passagens.....	8
2.6 Funções - Estatísticas.....	9
3. Conclusão.....	10
4. Bibliografia.....	11

1. Introdução

1.1. Tema e descrição do projeto

Sistema de gestão de passagem de veículos na A28

Este trabalho tem como objetivo apresentar em código-fonte de aplicação C, um sistema que simula a gestão das cobranças nos pórticos de portagens da A28.

No âmbito da disciplina de Algoritmos e Estrutura de Dados, este projeto tem como intuito aprofundar critérios lógicos essenciais relacionados à implementação de código e sua sintaxe, proporcionando uma leitura e compreensão abrangentes.

A investigação foi realizada para assegurar que o modelo desenvolvido para este sistema alcance uma amplitude significativa de elementos estruturais (p.ex: identificadores), garantindo assim uma funcionalidade mais sofisticada e sua composição. Isto leva, a uma aplicação de conceitos fundamentais, visando não apenas prática, mas também uma compreensão mais profunda no desenvolvimento e construção de um sistema.

- **Concepção do programa::**

Este sistema aplicativo permite definir e gerir a quantidade de veículos e verba gerado pela A28 em quatro pórticos diferentes. Onde cada pórtico terá um “X” número de identificação e uma tabela de preços consoante o tipo de classe a que o veículo pertence. Esta aplicação dispõe de outras mais diversas funcionalidades, de forma a permitir ao utilizador criar, adicionar e/ou remover.

O sistema foi sempre submetido a vários testes de forma a cumprir com as exigências do utilizador, proporcionando uma flexibilidade e adaptabilidade de gestão.

Portanto, este projeto destaca-se pela capacidade de interpretação, sintaxe e premissa na criação de sistemas na linguagem C, como também, práticas de conceitos fundamentais em Algoritmos e Estrutura de Dados ao desenvolvimento desta mesma.

- **Agentes relevantes:**

- Otimização da eficiência na cobrança de portagens, onde a implementação de um sistema eficaz de gestão na passagem de veículos visa otimizar esta mesma. Isso não apenas agrega valor ao processo de arrecadação da verba.
- Capacidade de monitorizar e gerir o fluxo de veículos promovendo a segurança e a eficiência na utilização da infraestrutura rodoviária.
- Contribuição vital de recursos para a manutenção e desenvolvimento contínuo da infraestrutura rodoviária. Um sistema eficiente de gestão não só facilita a arrecadação de recursos necessários, mas também proporciona transparência e confiabilidade nos processos financeiros associados à manutenção das estradas.
- Modernização e adaptação tecnológica de forma a melhorar a experiência e necessidades operacionais, proporcionando a eficiência e precisão na gestão dos veículos.

2. Desenvolvimento

2.1. Abordagem e definição do problema

O maior problema deste sistema é que ele pode chegar a ser complexo. Existem muitas opções disponíveis no qual pode ser difícil para o utilizador saber qual a opção escolher.

Para resolver estes problemas, o sistema terá que ser o mais simplificado possível de forma a não convergir entre funcionalidades.

A abordagem adotada no desenvolvimento do sistema baseou-se em uma estrutura modular, onde cada funcionalidade específica é tratada como um módulo independente. Isto permitirá a uma simples e fácil extensão do sistema com “updates” de novas funcionalidades ou modificação de módulos já existentes sem afetar o funcionamento global.

Ao longo do projeto foram adotados certos mecanismos e métodos, de modo a responder e atingir o funcionamento destas mesmas estruturas.

Neste projeto estão inseridas 18 funcionalidades que permitem a operacionalidade deste programa de gestão.

2.2. Plano de ação

- Avaliação de resultados e feedback entre alunos:
 - Iniciar uma fase de avaliação e feedback entre alunos de forma a promover uma cooperação mais sólida e eficiente prático na criação do sistema e identificação de áreas de melhoria.
 - Analisar cuidadosamente todo o feedback partilhado e identificar as estruturas essenciais face ao sistema requisitado pelo utilizador(Professor).
 - Priorizar todos os requisitos solicitados pelo utilizador.
- Otimização do desempenho:
 - Realização de vários testes em diferentes cenários de forma a identificar possíveis obstáculos.
 - Leitura e análise do código inserido (métodos, funções, estruturas, etc...), de modo a garantir um desempenho eficiente.
- Implementação de funcionalidades:
 - Avaliar a viabilidade de todas as funcionalidades com base nos requisitos do utilizador.
 - Priorizar as funcionalidades de acordo com o seu impacto e alinhamento com os objetivos do sistema.

2.3. Criação e identificação de estruturas.

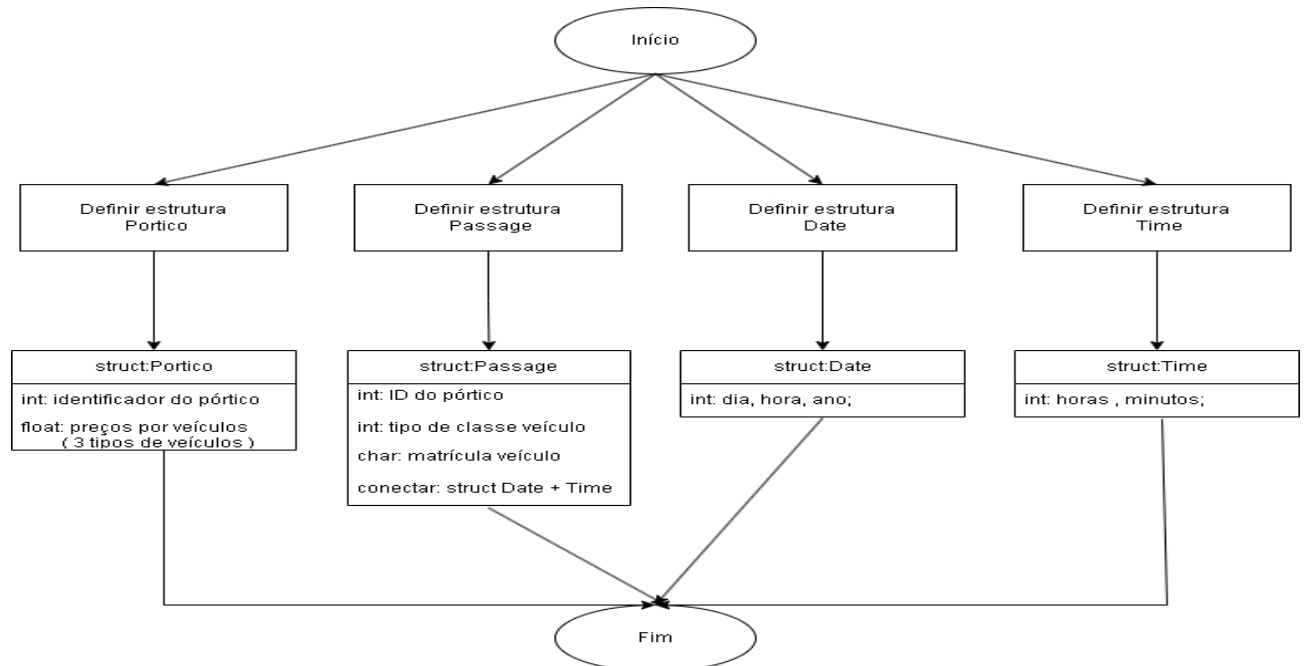


Fig.1 - Fluxograma sobre as estruturas criadas no sistema.

Estas estruturas são utilizadas para organizar e armazenar informações relacionadas a pórtilcos, passagens e dados temporais.

Cada estrutura declarada é lhe atribuída características e identificadores que facilitam a manipulação dessas informações no programa.

- Struct 'Portico':
 - Criação de um identificador único para cada pórtilco, permitindo a diferenciação entre eles. (1 a 4);
 - Criação de membros que irão armazenar os preços específicos para as diferentes classes de veículos em determinado pórtilco.
- Struct 'Passage':
 - O ID do pórtilco tem como objetivo identificar o pórtilco onde a passagem do veículo ocorreu, associando a passagem a um local específico.
 - Criar uma representação da classe do veículo, permitindo a categorização.
 - Criação de um membro string que armazena a matrícula de cada veículo, no qual identifica exclusivamente cada veículo.
 - Ligação da estrutura 'Date' no qual contém informações sobre a data da passagem.
 - Ligação da estrutura 'Time' no qual contém informações sobre o horário da passagem.
- Struct 'Date':
 - Criação de membros 'day', 'month', 'year', de modo a armazenar componentes de uma data específica, permitindo a representação completa de informações temporais.
- Struct 'Time':
 - Criação de membros 'hour', 'minutes', no qual armazenará componentes de um hora específica, oferecendo uma representação precisa do horário.

2.4. Funções - Gestão de pórticos

Envolve as funções:

- ‘addPortico’ : Adiciona um novo pórtico ao sistema;
- ‘removePortico’: Remove um pórtico existente do sistema;
- ‘listPorticos’: Lista de todos os pórticos existentes;
- ‘editPrices’: Edita os preços de um pórtico específico;
- ‘priceCheckForEachClass’: Verifica os preços dos pórticos para cada classe de veículo.

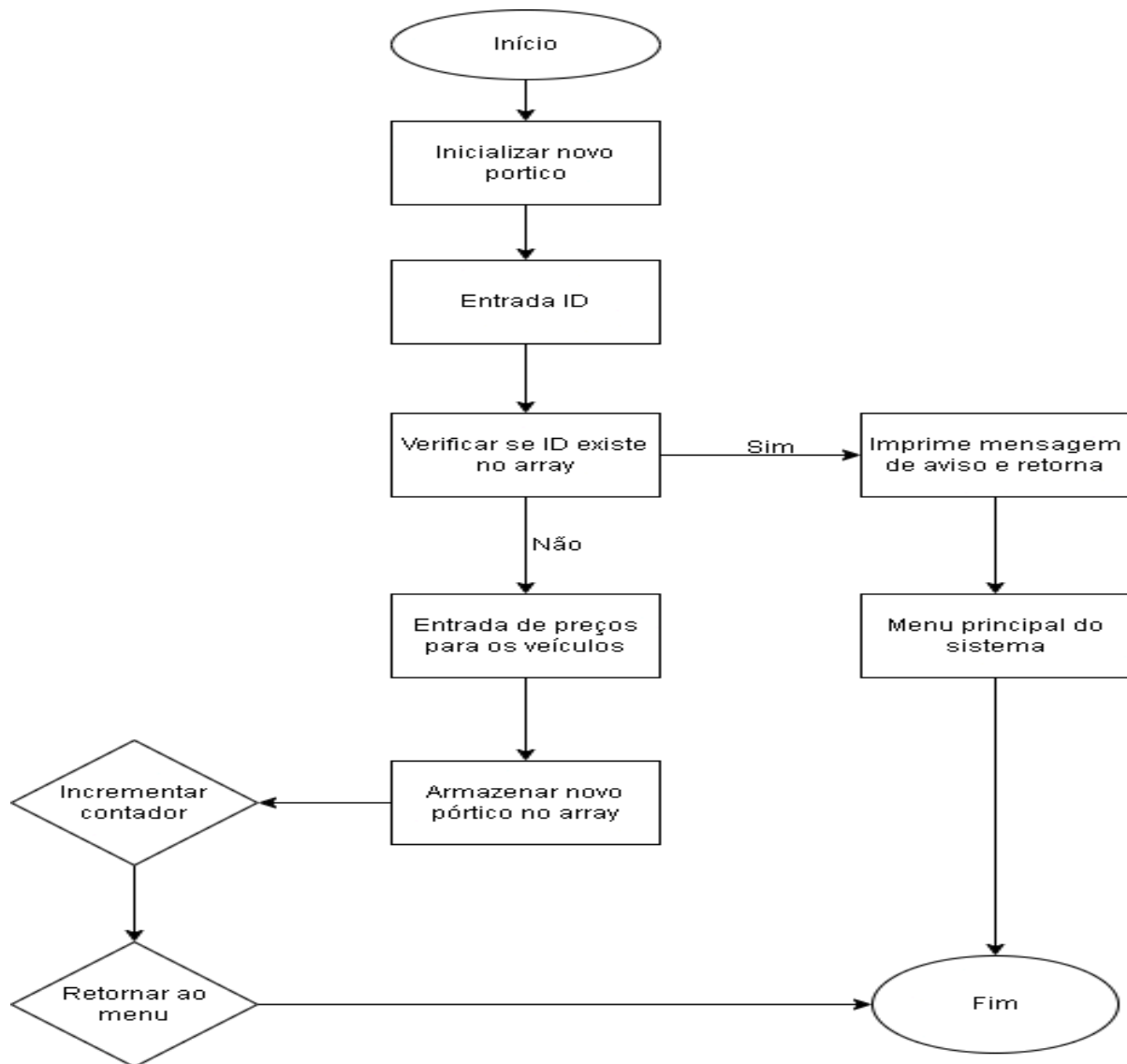


Fig.2 - Função ‘addPortico’.

A função “addPortico()” é uma função que permite ao utilizador adicionar um novo pórtico numa lista de pórticos e ajustar preços para cada tipo de classe de veículos.

Esta função permite verificar se o pórtico já existe na lista. Se existir, imprime uma mensagem de modo a informar ao utilizador e retorna à função ‘menu()’ - que representa o menu inicial do sistema. Caso o pórtico não exista, o sistema aceitará e adiciona à lista.

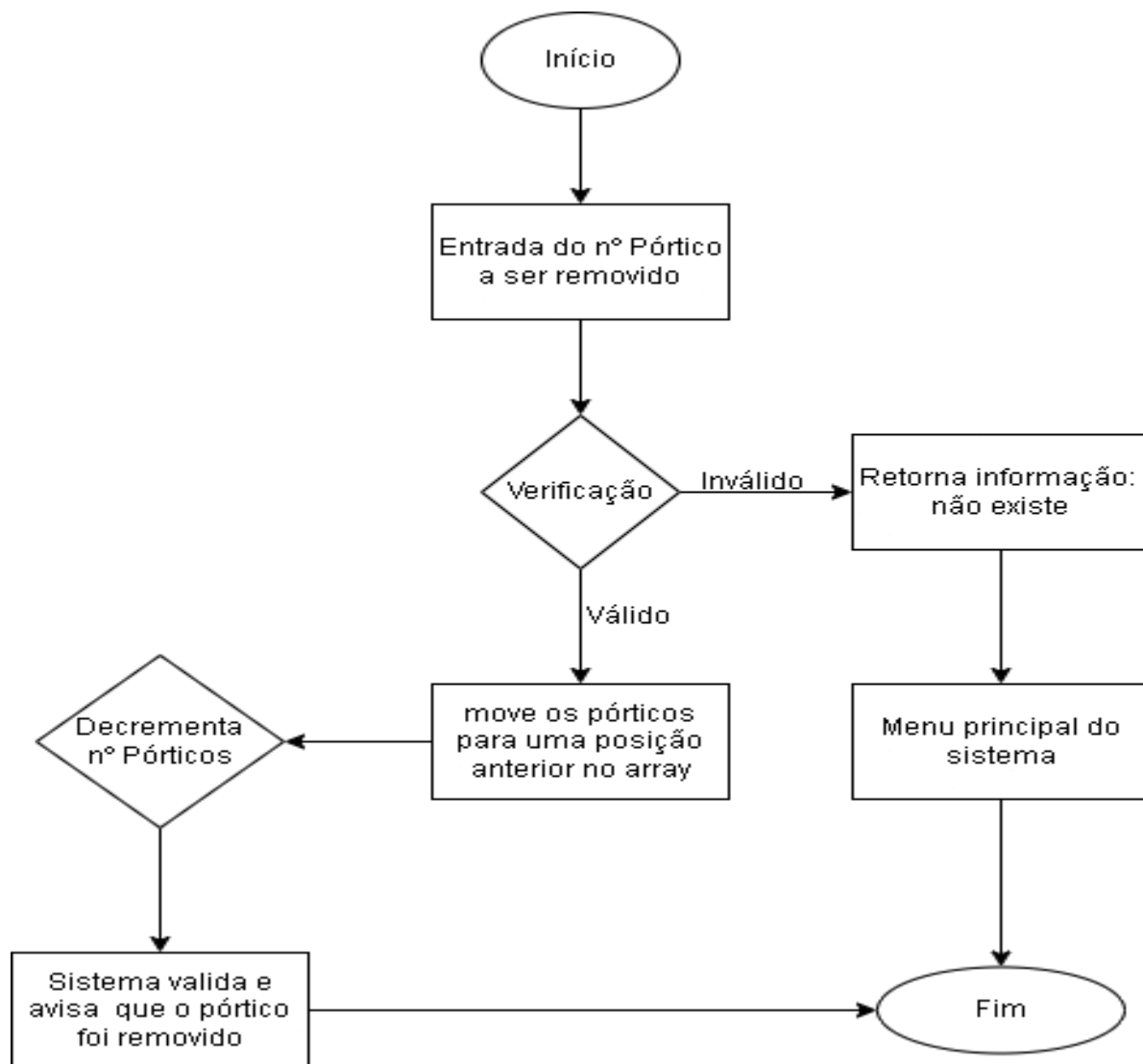


Fig.3 - Fluxograma função 'removePortico'.

A função 'removePortico' permite ao utilizador remover pórticos existentes do sistema de forma segura e eficiente. O número do pórtico a ser removido irá percorrer pela array de pórticos, encontrando o pórtico desejado e remove-o, ajustando o array para preencher o espaço vazio.

Esta função fornece feedback ao utilizador, indicando se o número do pórtico foi removido ou não. Após toda esta operação, o utilizador retornará ao menu principal.

- 'listPorticos': Exibe uma lista de todos os pórticos existentes no sistema, onde cada pórtico, mostra o seu número e preço de cada tipo de classe de veículos.
- 'editPrices': Permite modificar a tabela de preços para cada tipo de classe de veículos.
- 'priceCheckForEachClass': Mostra a tabela de preços para cada classe de veículos.

2.5. Funções - Gestão de Passagens

Envolve as seguintes funções:

- ‘insertPassage’: Insere uma passagem de um veículo no sistema;
- ‘listPassage’: Apresenta uma lista de todas as passagens registadas pelo sistema;
- ‘listPassageNumPortico’: Apresenta uma lista de todas as passagens registadas em cada pórtilco.

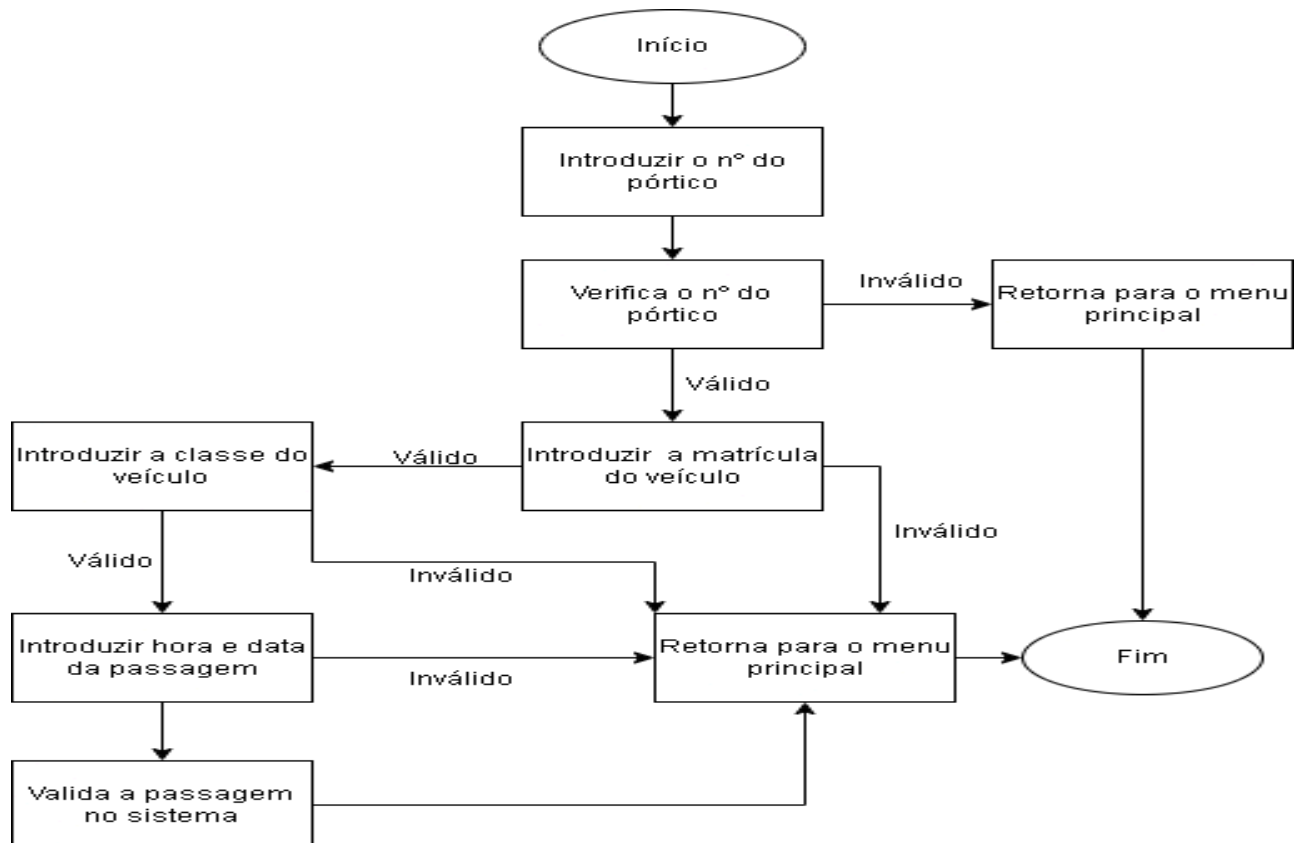


Fig.4 - Fluxograma função ‘insertPassage’.

A função ‘insertPassage’, é responsável por inserir uma nova passagem no sistema. Primeiramente solicita ao utilizador para identificar o pórtilco de passagem, depois o utilizador terá que preencher os campos como a matrícula do veículo, a classe do veículo, a data e a hora da passagem. Em seguida, a função valida os dados inseridos pelo utilizador de modo a garantir que eles estejam no formato correto e que a matrícula do veículo não esteja associada a outra passagem de outro pórtilco. Se os dados estiverem corretos, a função cria um objeto ‘passage’, e o adiciona ao vetor ‘passages’.

Por fim, a função incrementa o valor de ‘nrOfPassages’ para indicar que uma nova passagem foi adicionada ao sistema.

- ‘listPassage’: Mostra ao utilizador todas as passagens registadas, incluindo matrícula, data, hora e classe do veículo.
- ‘listPassageNumPortico’: Permite ao utilizador escolher qualquer pórtilco para verificar todas as matrículas dos veículos que passaram por “X” pórtilco.

2.6. Funções - Estatísticas

Envolve as seguintes funções:

- ‘RendimentoPorClasse’: Calcula o rendimento diário por classe de veículo;
- ‘averagePassagesPerPortico’: Calcula a média de passagens por dia por pórtilho;
- ‘totalTráfegoPorPortico’: Calcula o tráfego total (passagens) por pórtilho;
- ‘totalPassagePorticos’: Calcula o número total de passagens registradas no sistema;
- ‘trafegoDiariaPorClasse’: Calcula o tráfego diário (passagens) por classe de veículo;
- ‘PassagensVeiculoPorPortico’: Calcula o número de passagens realizadas por um veículo específico em um pórtilho específico;
- ‘RendimentoTotal’: Calcula o rendimento total do sistema;
- ‘gastoVeiculoAPorticos’: Calcula o gasto total de um veículo específico em todos os pórticos;
- ‘rendimentoDiarioPorPortico’: Calcula o rendimento diário por pórtilho;
- ‘porticoMaiorTráfego’: Identifica o pórtilho com o tráfego mais alto (número de passagens).

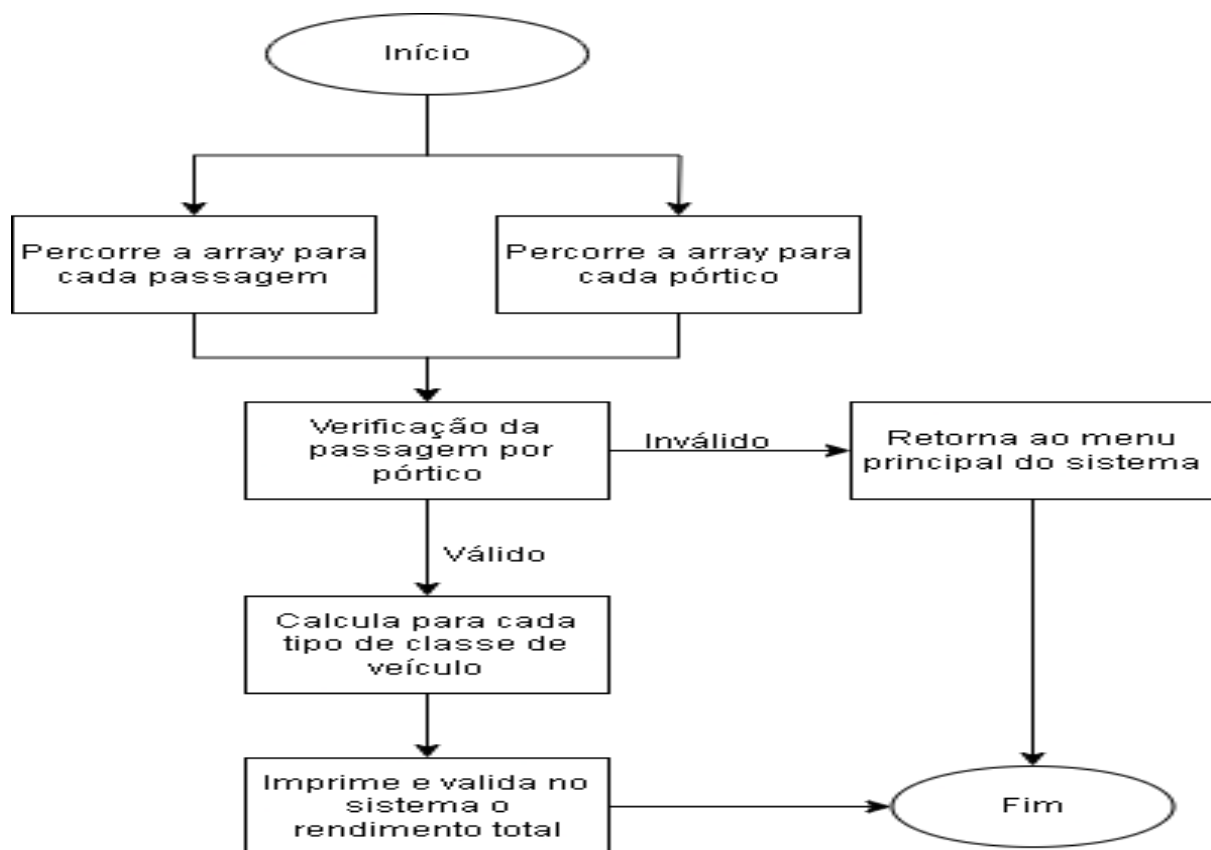


Fig. 5 - Fluxograma - Função ‘RendimentoTotal’.

A função ‘RendimentoTotal’ calcula o rendimento total de um conjunto de passagens por cada pórtilho. Para isso, ela percorre as passagens e, para cada passagem, verifica em qual pórtilho ela passou. Em seguida, calcula o preço correspondente à classe do veículo e adiciona ao conjunto do rendimento total.

A função verifica e impede caso algum dos campos - tipo de classe e/ou preço - sejam inválidos.

3. Conclusão

Este sistema de gestão de pórtricos e passagens na A28 oferece uma solução abrangente para monitorizar e analisar o tráfego em diferentes pórtricos. As funcionalidades de gestão de pórtricos permitem adicionar, remover e editar informações, enquanto as de gestão de passagens fornecem um registo detalhado de todas as passagens. As estatísticas representam “insights” sobre o desempenho do sistema e o comportamento do tráfego.

A capacidade de calcular rendimentos diários, médias, totais e específicos por classe de veículo e pórtrico fornece ao utilizador uma visão mais holística do sistema. A identificação do pórtrico com o maior tráfego é útil para otimizar operações. Além disso, a análise individual de passagens e gastos por veículo oferece uma perspetiva detalhada.

No geral, este sistema fornece ferramentas capazes de gerir estes pórtricos, análise de passagens e manipulação de dados. A implementação dessas funcionalidades contribui para um controle mais eficaz e uma compreensão aprofundada do sistema como um todo.

Durante o desenvolvimento deste sistema, um dos desafios encontrados foi a complexidade da sintaxe e a criação lógica de determinadas funções. A complexidade surgiu ao lidar com múltiplas funcionalidades e interações entre diferentes partes do programa. De forma a superar este desafio, os alunos decidiram abordar todos os métodos de forma mais modular, dividindo o código em funções mais pequenas e focadas, no qual facilitou a compreensão e manutenção do código.

Além disso, uma possível oportunidade de melhoria seria a implementação de um menu de login. A introdução de autenticação e perfis de utilizador incentiva a resolução da questão da segurança e ajuda a personalizar as funcionalidades disponíveis com base nas permissões atribuídas de cada utilizador. Isso não apenas aprimora a segurança do sistema, mas também fornece uma experiência mais personalizada aos utilizadores, adaptando-se às suas necessidades específicas.

Melhoria da interface do sistema é outro aspecto a ser considerado. Realizar ajustes na interface pode melhorar a sua eficiência, segurança e uso do sistema. Contribuiria para uma melhoria de desempenho e manutenção do programa.

4. Bibliografia

Website forum stackoverflow. “Funções Estáticas”. 2017

<https://pt.stackoverflow.com/questions/171410/fun%C3%A7%C3%B5es-est%C3%A1ticas-em-c/>

OpenAI. “ChatGPT”. <https://chat.openai.com/>

w3schools. “C Language”. <https://www.w3schools.com/c/index.php>

Via Verde. “Classes de veículos”. <https://www.viaverde.pt/particulares/ferramentas/classe-veiculos>

IMTT. “Veículos”. <https://www.imt-ip.pt/sites/IMTT/Portugues/Veiculos/Paginas/Veiculos.aspx>