

## Enunciado do Trabalho Prático n.º 1

### Cartão de Passageiro

16 de Março de 2023

## 1. Introdução

O objectivo deste trabalho é implementar um sistema de gestão de cartões online para passageiros da CP, com atribuição de pontos à semelhança dos cartões associados aos postos de abastecimento de combustível. Este documento descreve as regras a aplicar no desenvolvimento do programa informático.

## 2. Descrição do sistema de cartões online

O sistema gere informação referente a passageiros da CP registados online (que podem ser muitos milhares) e a estações de caminho de ferro aderentes. A criação de um cartão de pontos faz-se por registo do respectivo passageiro, processo esse que passa pela introdução do seu endereço de email.

Cada estação, caracterizada por um código único, tem um nome (não necessariamente único) e pertence a uma dada localidade (com nome único). Cada localidade pode ter várias estações.

O número inteiro de quilómetros de uma viagem efectuada por um passageiro é convertido directamente em pontos a acumular no seu cartão. Para além dos pontos, o cartão também retém o código da estação destino da viagem mais recente.

Para promover as suas localidades, os órgãos autárquicos patrocinam a existência de brindes associados a cada estação, que podem ser oferecidos aos passageiros a troco de um certo número de pontos subtraídos aos acumulados nos cartões. Cada estação tem um determinado tipo de brinde, podendo um passageiro escolher, de cada vez, um número arbitrário de brindes limitado ao *stock* existente e ao número de pontos no cartão. Os brindes escolhidos por um passageiro correspondem a 500 pontos por unidade se a viagem mais recente desse passageiro tiver sido para uma localidade diferente da que está a patrocinar aqueles brindes. Se, pelo contrário, a viagem mais recente tiver sido para a localidade que patrocina os brindes escolhidos, cada brinde valerá 300 ou 400 pontos, consoante a viagem registada no cartão do passageiro tenha sido para a estação com os brindes pretendidos ou para qualquer outra dessa mesma localidade.

## 3. Implementação

O objectivo deste trabalho é desenvolver um programa em C que implemente o sistema de gestão de cartões de passageiro.

### 3.1 Informação armazenada

Todos os dados associados ao sistema devem ser guardados num único ficheiro. Assim sendo, de cada vez que o programa é executado deve-se ler o ficheiro que contém essa informação, caso exista. Neste último caso, admite-se que o ficheiro está sempre sintacticamente correcto. Imediatamente antes do programa terminar, é necessário actualizar-se a informação no ficheiro. De notar que este ficheiro só é lido no início do programa e escrito no fim do mesmo programa. Não se deve, portanto, consultar ou actualizar o ficheiro em nenhum outro estado do programa.

### 3.2 Interacção com o utilizador

O programa deve permitir fazer a gestão dos passageiros e das estações da CP aderentes ao sistema de cartões. A interacção com o utilizador é realizada através de linha de comandos (interpretador de comandos). Caso o utilizador mande executar um comando não existente, o programa deve escrever a seguinte mensagem:

- `Comando invalido.`

Caso o utilizador mande executar um comando existente que necessita de informação (parâmetros do comando) e indique incorrectamente esses dados, o programa deve escrever a seguinte mensagem:

- `Dados invalidos.`

Nenhum parâmetro poderá conter espaços em branco, nem estes podem figurar antes do nome de um comando.

Caso o utilizador mande executar um comando existente, mas o programa não consiga responder por falta de recursos (por exemplo, memória insuficiente), o programa deve escrever a seguinte mensagem:

- `Inexistencia de recursos informaticos.`

### 3.3 Lista de Comandos

No caso da execução de um determinado comando não ter sucesso, apenas uma mensagem de erro deverá ser apresentada ao utilizador. Para os comandos em que possa haver mais do que uma causa de insucesso, nas descrições que se seguem as respostas possíveis encontram-se ordenadas por prioridade decrescente.

#### Comando RP

##### Linha de comando:

RP <emailPassageiro> <nomePassageiro> ↵

##### Descrição:

Regista o passageiro de nome <nomePassageiro> e cujo código é o endereço de email <emailPassageiro>, criando-lhe um cartão de pontos online.

**Respostas possíveis:****Caso a operação não tenha sucesso**

- Passageiro ja anteriormente registado.

**Caso a operação tenha sucesso**

- Registo de passageiro bem sucedido.

**Comando RE****Linha de comando:**

RE <códigoEstação> <nomeEstação> <localidade> <stockDeBrindes> ↵

**Descrição:**

Regista a estação da CP com código <códigoEstação> e nome <nomeEstação>, pertencente à localidade <localidade> e possuindo o número <stockDeBrindes> de brindes disponíveis.

**Respostas possíveis:****Caso a operação não tenha sucesso**

- Estacao ja anteriormente registada.

**Caso a operação tenha sucesso**

- Registo de estacao bem sucedido.

**Comando IB****Linha de comando:**

IB <códigoEstação> <unidadesAdicionadas> ↵

**Descrição:**

Incrementa de <unidadesAdicionadas> o número de brindes disponível na estação identificada por <códigoEstação>. O valor do incremento deve ser um inteiro maior do que zero.

**Respostas possíveis:****Caso a operação não tenha sucesso**

- Estacao desconhecida.

**Caso a operação tenha sucesso**

- Actualizacao de brindes bem sucedida.

**Comando RV****Linha de comando:**

RV <emailPassageiro> <códigoEstaçãoDestino> <pontos> ↵

**Descrição:**

Regista, no cartão do passageiro identificado por <emailPassageiro>, o código <códigoEstaçãoDestino> da estação destino referente à viagem mais recente, acumulando também o respectivo número <pontos> de pontos, sempre maior do que zero.

**Respostas possíveis:**

**Caso a operação não tenha sucesso**

- Passageiro desconhecido.
- Estacao desconhecida.

**Caso a operação tenha sucesso**

- Registo de viagem bem sucedido.

**Comando EB**

**Linha de comando:**

EB <emailPassageiro> <códigoEstação> <quantidadePretendida> ↵

**Descrição:**

Escolha do número <quantidadePretendida> de brindes da estação <códigoEstação> pelo passageiro <emailPassageiro>. O valor <quantidadePretendida> deverá ser um inteiro maior que zero. Os pontos correspondentes aos brindes escolhidos deverão ser descontados no cartão do passageiro, não sendo permitidos saldos negativos.

**Respostas possíveis:**

**Caso a operação não tenha sucesso:**

- Passageiro desconhecido.
- Estacao desconhecida.
- Quantidade indisponivel de brindes na estacao.
- Quantidade insuficiente de pontos no cartao.

**Caso a operação tenha sucesso**

- Escolha de brindes bem sucedida.

**Comando CE**

**Linha de comando: CE**

<códigoEstação> ↵

**Descrição:**

Processo de consulta da informação associada à estação <códigoEstação>.

**Respostas possíveis:**

**Caso a operação não tenha sucesso**

- Estacao desconhecida.

#### **Caso a operação tenha sucesso**

<Código da Estação> ↵

<Nome da Localidade> - <Nome da Estação> ↵

Numero de brindes = <Quantidade disponível de Brindes> ↵

### **Comando LP**

#### **Linha de comando:**

LP ↵

#### **Descrição:**

Listagem de todos os passageiros detentores de cartão de pontos. Figurarão nessa listagem, para cada passageiro, o seu código e o seu nome, bem como o número de pontos acumulados no cartão. Para além disso, se for um passageiro com o mínimo de uma viagem já efectuada, também serão listados o código da estação da viagem mais recente e o nome da respectiva localidade.

#### **Respostas possíveis:**

##### **Caso a operação não tenha sucesso**

- Nao existem passageiros registados.

##### **Caso a operação tenha sucesso** (sendo condicionais os 2 últimos valores de cada linha)

<email1> <Nome do Passageiro1> <Pontos1> <Código Estação1> <Localidade1> ↵

<email2> <Nome do Passageiro2> <Pontos2> <Código Estação2> <Localidade2> ↵

...

<emailN> <Nome do PassageiroN> <PontosN> <Código EstaçãoN> <LocalidadeN> ↵

### **Comando TB**

#### **Linha de comando:**

TB ↵

#### **Descrição:**

Listagem com a(s) localidade(s) que apresentar(em) maior número de brindes distribuídos (*Top* de Brindes). Esse número também deverá figurar na listagem. De notar que o número de brindes distribuídos por cada localidade é obtido tendo em conta todas as estações que lhe pertencem.

Se houver mais do que uma localidade nas condições indicadas, a listagem ficará ordenada da que mais recentemente distribuiu brindes (através das suas estações) para a que os distribuiu há mais tempo. Por exemplo, se ambas as localidades A e B, de entre todas as existentes, alcançarem um máximo de 40 brindes distribuídos, a localidade A deverá ser listada antes da B se tiver havido uma estação de A que tenha distribuído brindes mais recentemente que qualquer estação de B.

**Respostas possíveis:****Caso a operação não tenha sucesso**

- Não existem estações registadas.

**Caso a operação tenha sucesso, mas não tendo havido ainda distribuição de brindes**

- Ainda não foram distribuídos quaisquer brindes.

**Caso a operação tenha sucesso, já tendo havido distribuição de brindes**

Numero de Brindes = *<Valor do maior número de Brindes por Localidade>* ↵

*<LocalidadeI>* ↵

...

*<LocalidadeN>* ↵

**Comando A****Linha de comando:**

A ↵

**Descrição:**

Pedido de ajuda, que produzirá uma listagem de todos os comandos possíveis e dos respectivos parâmetros. Cada comando não deverá ocupar mais do que uma linha no ecrã.

**Comando S****Linha de comando:**

S ↵

**Descrição:**

Comando de saída do programa, terminando a execução deste.

## 4. Entrega do trabalho

O trabalho deverá ser enviado até ao dia 10 de Abril de 2023

## 5. Critérios de avaliação

A avaliação do trabalho será feita de acordo com os seguintes princípios:

- Estruturação: o programa deve estar estruturado em tipos abstractos de dados considerados necessários e convenientes para a resolução do problema.
- Correção: o programa deve executar todas as funcionalidades, tal como pedido.
- Legibilidade e documentação: o código deve ser escrito, formatado e comentado de acordo com o standard de programação definido para a disciplina.