

Instituto Superior Técnico
DEEC
Programação 2018/2019 – 1º S
2º Trabalho (versão final.)

1 Objectivo

Neste trabalho pretende-se desenvolver um simulador de redes ferroviárias (jogo simplificado) utilizando a linguagem de programação C. Com essa finalidade é necessário aplicar conceitos de organização de dados, definindo novos tipos baseados em estruturas de dados e aplicando conceitos de memória dinâmica para guardar a informação do estado de execução do simulador.

2 Simulador

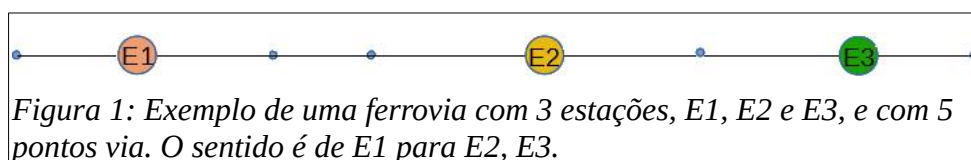
O simulador deve ser desenvolvido de modo a ser possível descrever:

1. Um conjunto de **linhas ferroviárias** que são representadas numa janela gráfica.
2. Um conjunto de **comboios** que circulam nas linhas ferroviárias.
3. Realização de acções para gerir as linhas e para movimentar os comboios.

A **configuração** do simulador/aplicação é realizada através de um **ficheiro de texto** que contém as instruções para definir o conjunto de linhas ferroviárias e o conjunto de comboios. O nome do ficheiro não é fixo.

3 Linhas ferroviárias

Uma linha ferroviária é constituída por um identificador (conjunto de 4 caracteres úteis) e um conjunto de **pontos**. Os pontos são classificados em pontos de via ou pontos estação, ver a descrição na figura 1. O sentido de circulação numa ferrovia é único e é definido pela ordem com que os pontos da ferrovia são especificados no ficheiro de configuração da aplicação.



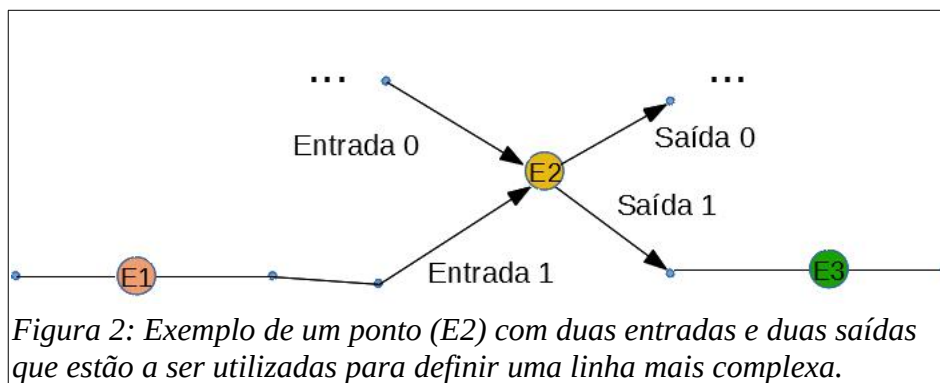
Um ponto é definido com base na informação seguinte:

1. **Identificador de ponto:** Conjunto de 4 caracteres úteis.
2. **Coordenadas** gráficas x e y (píxeis) para posicionamento do ponto na janela gráfica.
3. **Cor** do ponto. A cor dos pontos de via pode ser utilizada para facilitar a identificação das ferrovias. Sugestão: Os pontos de via da mesma ferrovia devem ter a mesma cor.
4. **Tipo de ponto:** VIA-Via ou EST- Estação. Atenção: De modo a não sobrecarregar o

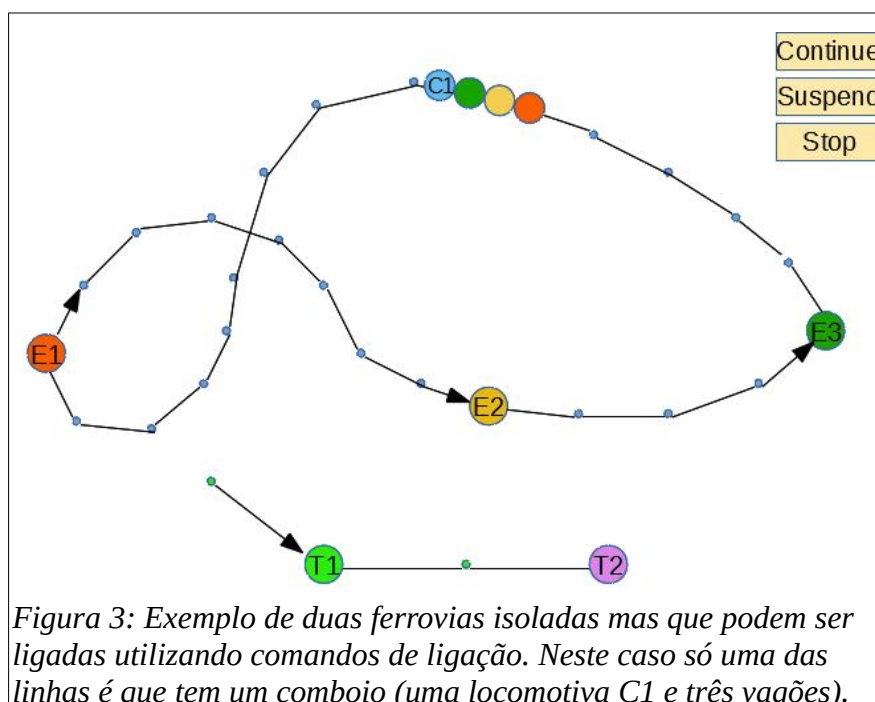
desenho gráfico as identificações dos pontos de via não devem ser representadas na janela gráfica.

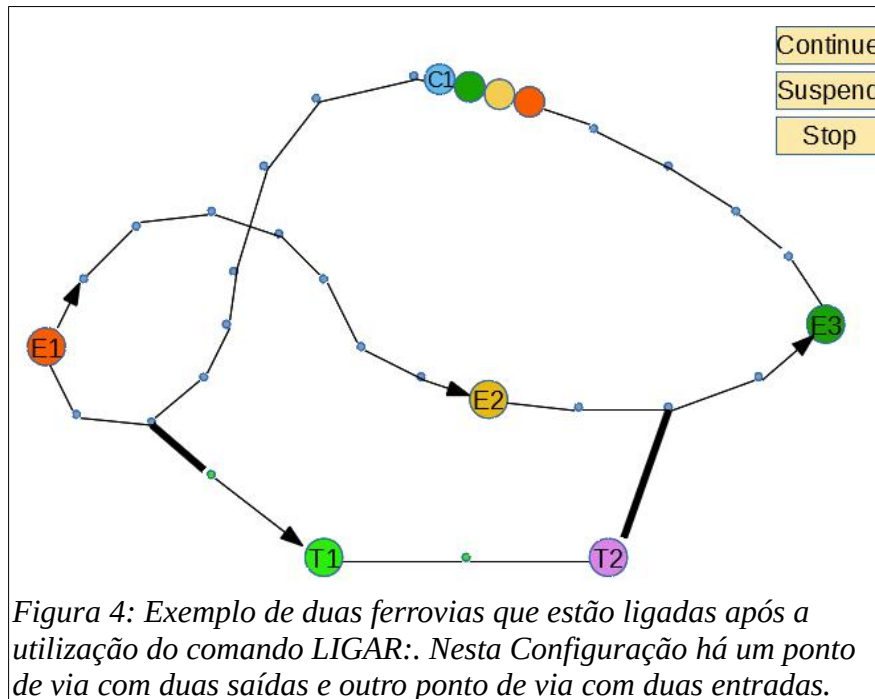
5. **Dimensão** (píxeis) do ponto: Os pontos de via têm dimensões menores do que as estações. Todos os pontos de via têm a mesma dimensão. Todos as estações tem a mesma dimensão.
6. Número de **vias de entrada** num ponto: máximo 2 .Ver a figura 2.
7. Números de **vias de saída** de um ponto: máximo 2. Ver a figura 2.

Na figura 2 são exemplificadas as entradas e as saídas num ponto da ferrovia que pode ser um ponto de via ou um ponto estação. As setas são apresentadas para exemplificar o sentido de cada uma das linhas. A numeração das entradas e das saídas é feita em relação ao sentido da via.



Na figura 3 são apresentadas duas linhas que estão separadas. No entanto as duas ferrovias podem ser ligadas (figura 4) utilizando instruções de ligação (LIGAR:) que devem ser colocadas no ficheiro de configuração, depois das instruções de definição das linhas.





4 Comboios

Um comboio é definido pela informação seguinte (ver as figuras 3 e 4):

1. **Identificador:** Conjunto de 2 caracteres úteis. Deve ser único.
2. **Dimensão do comboio:** 4 bolas.
3. **Dimensão das bolas:** Raio em píxeis.
4. **Cor:** A cor do comboio é a cor da locomotiva. As carruagens/vagões também têm cor. A cor cinzenta indica que a carruagem está vazia. Outras cores indicam que os vagões estão ocupadas com “produtos” os quais devem ser entregues nas estações que tenham a cor do vagão. Quando o vagão passa pela estação com a mesma cor, a cor do vagão passa para a cor cinzenta. As cores dos vagões são geradas de forma aleatória utilizando a função **rand()**. **Identificadores das cores,** VERMELHO, ROXO, AZUL, CYAN, VERDE, AMARELO, CASTANHO, PRETO, BRANCO (e CINZENTO).
5. **Ponto de início de marcha:** O local de colocação para o início de marcha é indicado no ficheiro de configuração através do nome da ferrovia e do ponto na ferrovia. Também deve ser indicado pelo utilizador quando selecciona a opção 5 na lista de opções (ver secção 5).
6. **Geração de comboios:** Um comboio pode ser gerado através da lista de opções (ver a secção 5), o programa deve pedir os dados relevantes para gerar um comboio. Também pode ser gerado de forma automática com a informação que consta no ficheiro de configuração, comando COMBOIO:..

5 Funcionalidades do programa

O programa deve funcionar de acordo com as regras seguintes,

1. Deve ler o nome do ficheiro de configuração através da linha de comando,

Exemplo: **sim_linhas ficheiro.txt**

2. O programa deve apresentar no ecrã (em modo texto) uma lista de opções como se exemplifica a seguir, e que serve de guia ao utilizador:

Lista de opções:

0 - Terminar o programa

1 - Mostrar a informação de uma ferrovia

2 - Eliminar uma ferrovia

3 - Mostrar a informação de um comboio

4 - Eliminar um comboio

5 - Criar um comboio

6 - Simulação dos comboios

Selecione a opção:

O utilizador selecciona uma das opções pressionando uma das teclas **0, 1, ..., 6** do teclado (seguido de Enter). O programa deve ler a opção e deve executar o trabalho correspondente. Após o processamento das opções **1, ..., 6** o programa deve apresentar novamente a lista de opções. A opção **0** corresponde a terminar o programa. Nota, a operação de eliminação não afecta o conteúdo do ficheiro de configuração. O ficheiro de configuração deve ser construído e modificado utilizando um editor de texto.

3. Na opção 6, deve ser possível suspender a simulação, regressando ao menu para executar operações (1 a 5), podendo depois retomar a sua execução utilizando o rato na janela gráfica. Ver a figura 3 ou 4. A simulação da velocidade do comboio deve ser especificada pelo programador.
4. Sempre que um comboio passe por uma estação e a cor de um vagão seja igual à cor da estação então a cor do vagão deve passar para a cor cinzenta.
5. Um comboio deve ser eliminado da simulação se chegar ao fim de uma linha, mas desde que o ponto final da linha não esteja ligado a outro ponto.
6. Deve ser possível comandar os comboios, parar individualmente um comboio e retomar a marcha. Para isso o utilizador deve clicar com o rato na (bola que representa a) locomotiva.
7. Deve ser possível efectuar a passagem de comboios entre as diversas linhas (ferrovias) que estejam ligadas com pontos de via ou estações com várias linhas de saída/entrada.

8. Deve ser detectada a possibilidade de ocorrência de colisões entre comboios, devendo ser executada a suspensão da marchas deles. Os comboios que estejam parados devem começar a piscar a cor da locomotiva. A ordem de marcha é executada clicando na locomotiva.
9. Deve ser construído um mecanismo de interacção com o programa através da janela gráfica para efectuar a selecção da via de saída (activação) nos pontos da linha que têm mais de uma via de saída. Em alternativa (mas com uma penalização na nota), este requisito pode ser realizado através da lista de opções em modo texto.

6 Organização de dados e do código

Este trabalho deve ser realizado utilizando tipos novos estruturados e o conceito de listas dinâmicas.

O código do programa deve estar organizado em módulos/ficheiros separados e deve ser aplicado o processo de compilação de módulos separados, utilizando a ferramenta **make**.

1. As funções devem estar documentadas e as instruções devem estar bem alinhadas.
2. Uma função não deve ter mais de 50 a 60 linhas, tamanho da fonte 12pt.
3. Uma função deve ter no máximo 5 argumentos.

Nota: O trabalho pode ser desenvolvido no **codeblocks**, sendo para isso necessário configurá-lo para ter acesso às bibliotecas gráficas do SDL2.

7 Organização do ficheiro de configuração

O ficheiro de configuração da aplicação é um ficheiro de texto em que a informação está organizada por linhas.

As linhas que no início tenham o carácter % devem ser ignoradas na leitura do ficheiro, funcionando como comentários. As linhas vazias (iniciadas com o carácter '\n', ou tendo somente espaços) também devem ser ignoradas.

No início do ficheiro são especificadas a largura e a altura da janela gráfica em píxeis. A linha está identificada com a palavra-chave **JANELA:**.

A definição de uma ferrovia começa com a palavra-chave **LINHA:** e termina com a palavra-chave **FIM_DE_LINHA:**. Nas linhas onde estão estas palavras também é indicado o nome da ferrovia (exemplo **LINHA: L1**, consultar o exemplo da organização do ficheiro). As linhas que estão entre a linha com **LINHA:** e a linha com **FIM_DE_LINHA:** descrevem pontos da via férrea, exemplo:

W1 20 30 AZUL EST

W1 indica o nome do ponto, 20 e 30 representam respectivamente as coordenadas x e y em píxeis, AZUL indica a cor do ponto, EST é um código para indicar que o ponto é um ponto de estação na via férrea.

O ficheiro pode conter informação de diversas ferrovias. As ferrovias podem ser ligadas utilizando o comando **LIGAR:** exemplo

LIGAR: COA C20 LAM P1

O primeiro par formado por COA C20 indica o local de saída na ferrovia COA no ponto C20. O segundo par formado por LAM P1 indica a entrada na ferrovia LAM no ponto P1.

No ficheiro também pode existir informação para descrever comboios, exemplo

COMBOIO: R1 3 VERMELHO L1 W1 5

A palavra-chave COMBOIO: indica que a linha de texto contém informação para descrever um comboio. R1 indica o nome do comboio, o número 3 (raio) define o tamanho em píxeis para cada uma das quatro bolas que representam o comboio, o código **VERMELHO** indica a cor da locomotiva e L1 W1 indica o nome da ferrovia e o ponto na ferrovia de onde o comboio começa o seu movimento. O número seguinte (5) indica o número de viagens (serviços) que o comboio vai realizar. Sempre que um comboio alcance um ponto terminal de uma linha completa um serviço e deve reaparecer novamente no ponto inicial de marcha.

Exemplo da organização do ficheiro de configuração (config_model.txt):

% JANELA: palavra-chave seguido das dimensões x y da janela gráfica

JANELA: 800 600

% LINHA: - palavra-chave seguido do identificador

LINHA: COA

% Informacao dos pontos, ordem implica sentido de ligação

C1 20 30 AZUL VIA

C2 60 80 AZUL EST

C3 90 90 AZUL VIA

...

C19 100 120 VERMELHO EST

C20 100 130 AZUL VIA

FIM_DE_LINHA: COA

% LINHA: - palavra-chave seguido do identificador

LINHA: LAM

P1 200 200 VERDE EST

P2 200 210 AMARELO VIA

...

P10 300 300 AMARELO VIA

FIM_DE_LINHA: LAM

%LIGAR: palavra-chave para ligar linhas

% (saída de uma linha -> entrada na outra linha)

LIGAR: COA C20 LAM P1

LIGAR LAM P10 COA C1

% COMBOIO: Palavra chave para identificar um comboio

COMBOIO: C1 3 VERMELHO COA C1 10

COMBOIO: C2 3 AZUL LAM P1 5

...

8 Desenvolvimento da aplicação e utilização da biblioteca gráfica

No desenvolvimento da aplicação é obrigatório a utilização da linguagem de programação C, norma C99. O acesso às funcionalidades gráficas é realizado utilizando a **biblioteca gráfica SDL2** que está instalada em todos os computadores do laboratório. Na página da disciplina e nas aulas de laboratório será exemplificada a utilização da biblioteca gráfica.

9 Entrega e avaliação

A aplicação será avaliada nos computadores do laboratório em ambiente (sistema operativo) Gnu-Linux. Os alunos devem testar a sua aplicação nos computadores de laboratório ANTES de realizarem o *upload* da versão final no sistema fénix.

Devem colocar num ficheiro zip o(s) ficheiro(s) do código fonte (ficheiros *.h e *.c) conjuntamente com o ficheiro de configuração que utilizou para desenvolver o programa. Não deve incluir o programa executável.

- **Data de entrega:** Finais de Dezembro, mas é necessário consultar a página da disciplina para verificar se há actualizações da data ou do enunciado do trabalho.

10 Dúvidas

As dúvidas podem ser esclarecidas no início e no fim das aulas, e no horário de esclarecimento de dúvidas.

Nota: Na definição inicial das funcionalidades de uma aplicação, existem elementos que não estão completamente especificados. Nessas situações, o programador pode especificar os elementos em falta utilizando para o efeito justificações lógicas.

Votos de um bom trabalho!