



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa



Gestão de tráfego ferroviário

Resolução de conflitos em linhas de baixo tráfego

José Pedro da Luz Bernardo de Figueiredo

Dissertação para obtenção do Grau de Mestre em

Engenharia Electrotécnica e de Computadores

Júri

Presidente: Prof. Francisco Miguel Prazeres Silva Garcia

Orientador: Prof. Carlos Filipe Gomes Bispo

Vogais: Prof. Paulo Fonseca Teixeira

Prof. David Martins de Matos

Novembro de 2007

Agradecimentos

Durante a elaboração desta tese foram várias as pessoas que contribuíram para que a levasse a bom termo, entre elas gostaria de salientar o meu orientador, Prof. Carlos Bispo, cujo apoio teve especial importância na formulação do projecto. A colaboração do Eng. João Piusa, que liderou a equipa da Alcatel e cuja orientação foi essencial no levantamento dos requisitos para o projecto. E o apoio da minha namorada e família que estiveram comigo durante toda a elaboração da tese e da dissertação.

Resumo

O objectivo deste trabalho consiste na criação de uma solução para a resolução de conflitos em linhas de baixo tráfego, designada por Problema de Encontro e Passagem – *Meet and Pass* na literatura de origem inglesa –, em particular para as situações de uma linha única entre estações. Os principais conflitos a prevenir e detectar neste género de linhas são a presença de dois comboios com sentidos opostos na mesma linha e a ultrapassagem de comboios. Estes conflitos derivam essencialmente do facto de existir apenas uma linha a separar as diversas estações e desta ter de servir ambos os sentidos.

A solução proposta, que surge na forma de um sistema de apoio à decisão, possibilita a realização de uma simulação dos horários de um conjunto de comboios e a previsão de possíveis conflitos.

A aplicação desenvolvida contém uma interface gráfica que permite ao utilizador interagir com o programa. É permitido ao utilizador introduzir alterações nos horários dos comboios, alterações essas que podem corresponder a conflitos ou a possíveis soluções. É ainda possível pedir um gráfico com a execução dos horários dos vários comboios, de forma a adquirir uma perspectiva global da situação.

O simulador desenvolvido funciona recorrendo a eventos discretos e como resultado final produz uma lista dos conflitos detectados.

O objectivo da aplicação desenvolvida, no âmbito da resolução do problema inicial, é ser utilizada por um gestor de tráfego ferroviário que, aliando a sua experiência às capacidades de previsão de conflitos do simulador, produza soluções mais completas e fiáveis para a resolução de conflitos.

Palavras-Chave

Comboio, conflito, sistema de apoio à decisão, horário, linha de baixo tráfego, encontro e passagem.

Abstract

The goal of this project is the development of a solution for the conflict resolution in lines of low traffic, designated as the Meet & Pass Problem, in particular for the situations of a single line between stations.

The main conflicts to prevent and to detect are the presence of two trains with opposing directions in the same line and the proximity between trains, when a fast train follows behind a slow train. These conflicts derive essentially from the fact that there is a single line to separate the stations and that this line has to be used by the trains in both directions.

The proposed solution, which appears in the form of a decision support system, makes possible the accomplishment of a simulation of the schedules of a set of trains and the forecast of possible conflicts.

The developed application contains a graphical interface that allows the user to interact with the program. The user can introduce alterations in the schedules of the trains. These alterations can correspond to conflicts or to possible solutions. It is also possible to ask for a graph with the execution of the train schedules, which give a global perspective of the situation. As a final result, the simulator produces a list of the detected conflicts.

The idea of the developed application, in the scope of the resolution of the initial problem, is to be used by a railroad traffic manager that, allying its experience to the conflicts forecasting capabilities of the simulator, producing more complete and efficient solutions for the conflict resolution.

Keywords

Conflict, decision support system, low traffic line, single line, train schedule, *Meet and Pass*.

ÍNDICE

Agradecimentos	1
Resumo	2
Abstract	3
Lista de figuras	5
Lista de Tabelas	6
1. Introdução	7
1.1 Descrição dos Capítulos	8
2. Enquadramento	10
2.1 – Problema do “Meet and Pass” e propagação de conflitos	11
2.2 – Soluções existentes	12
2.3 – Solução Proposta	13
2.4 – Análise comparativa da solução proposta com as soluções existentes	14
3. Texto Principal	15
3.1 Opções de projecto (nível global).....	17
3.2 Descrição Funcional da Aplicação	24
3.3 Descrição Técnica da Aplicação	38
3.3.1 Tecnologia Utilizada.....	38
3.3.2 Opções de Projecto	39
3.3.3 Principais entidades criadas.....	41
4. Conclusões Finais	44
4.1 Melhorias possíveis de introduzir numa futura versão da aplicação	44
4.2 Dificuldades durante o desenvolvimento	46
4.3 Trabalhos futuros	47
Anexo A - Manual de Utilização	49
A1. Ficheiros de Input	49
A2. Acções disponíveis	51
Anexo B – UML da aplicação	58

Lista de figuras

Figura 1 - Estrutura geral da aplicação.....	15
Figura 2 - Exemplo de utilização da aplicação.....	16
Figura 3 – Modelo da linha de comboio.....	17
Figura 4 – Ecrã principal da aplicação.....	20
Figura 5 – Detalhe das opções no menu da aplicação.....	21
Figura 6 – Detalhe da secção de comboios.....	21
Figura 7 – Detalhe da secção de Horários.....	22
Figura 8 – Detalhe da secção de relatório de simulação.....	22
Figura 9 – Exemplo de gráfico gerado.....	23
Figura 10 – Processo de interacção com a aplicação.....	25
Figura 11 – Processo geral de carregamento de dados.....	26
Figura 12 – Processo de carregamento de estações.....	27
Figura 13 – Processo de carregamento de Comboios.....	28
Figura 14 – Processo de carregamento de linhas.....	29
Figura 15 – Processo de carregamento de horários.....	30
Figura 16 – Processo de gravação de log.....	31
Figura 17 – Processo de gravação das alterações introduzidas nos horários.....	31
Figura 18 – Processo de criação de um gráfico dos horários dos vários comboios.....	32
Figura 19 - Processo de criação do gráfico do horários de um comboio.....	33
Figura 20 Detecção de conflitos do tipo Pass.....	34
Figura 21 – Processo geral do funcionamento do simulador.....	35
Figura 22 – Processo de tratamento de chegada de uma comboio a uma estação.....	36
Figura 23 – Processo de tratamento da partida de um comboio.....	37
Figura 24 – Ecrã de escolha de ficheiro.....	40
Figura 25 – Menu de carregamento de ficheiros.....	51
Figura 26 – Janela para procura de ficheiro.....	52
Figura 27 – Ecrã principal, selecção da área de comboios.....	53
Figura 28 - Ecrã principal, selecção da área de horários.....	54
Figura 29 - Ecrã principal, selecção das opções de gravação e exportação de alterações.....	55
Figura 30 - Ecrã principal, selecção da opção de iniciar simulação e simulation report.....	55
Figura 31 – Ecrã de simulation report, selecção de geração de gráficos.....	56
Figura 32 – Ecrã de simulation report, selecção de opção para exportação.....	57
Figura 33 - Diagrama UML da aplicação desenvolvida.....	58
Figura 34 – UML da classe Troco.....	59
Figura 35 – UML da classe Station.....	59
Figura 36 – UML da classe Simulator.....	59
Figura 37 – UML da classe Train.....	60
Figura 38 – UML das classes HorarioAux e TimeBlock.....	61
Figura 39 – UML da classe TimeBlockPlus.....	61
Figura 40 – UML da classe Segment.....	61
Figura 41 – UML da classe HorarioTrain.....	61
Figura 42 – UML da classe Log.....	62
Figura 43 – UML da classe Event.....	62
Figura 44 – UML da classe EventList.....	62
Figura 45 – UML da classe Interface.....	63
Figura 46 – UML da classe Dados.....	64
Figura 47 – UML da classe MathAdvanced.....	64

Lista de Tabelas

<i>Tabela 1 – Formato dos dados do ficheiro de estações</i>	49
<i>Tabela 2 – Formato dos dados do ficheiro de comboios</i>	49
<i>Tabela 3 – Formato dos dados do ficheiro de linhas</i>	50
<i>Tabela 4 – Formato de dados do ficheiro de horários</i>	51
<i>Tabela 5 – Mensagens de relatório de conflitos</i>	56

1. Introdução

O tema desta dissertação insere-se num dos grandes problemas do mundo ferroviário, desde o início da sua existência até aos dias de hoje, a gestão dos comboios. Em particular a gestão dos seus horários.

A vertente, da gestão de horários, abordada nesta tese foi a dos problemas de encontro e passagem ou *Meet and Pass*. A própria nomenclatura faz uma alusão aos problemas que ocorrem. O *Meet* consiste no encontro de dois comboios ou seja, nas diversas situações em que se tem comboios na mesma linha, em direcções opostas e que a dada altura terão de se cruzar. O *Pass* corresponde às situações em que existem dois comboios na mesma linha e no mesmo sentido, mas onde o comboio mais atrás é mais rápido e a dada altura terá de realizar uma ultrapassagem.

Um horário pré-estabelecido contempla logo à partida todos os pontos de encontro e passagem que serão necessários. Os problemas surgem derivados das constantes falhas e ocorrências imprevisíveis no decorrer do horário. Estas ocorrências estão tipicamente sob a forma de simples atrasos dos comboios nas partidas e nas chegadas, passando por problemas nas linhas físicas que possam condicionar ou impedir a circulação nas mesmas, ou ainda de avarias de comboios durante o seu percurso e que originem o corte da linha em questão. A ocorrência de qualquer um destes problemas origina mudanças nos horários dos comboios intervenientes, fazendo com que o plano pré-estabelecido deixe de garantir a ausência de conflitos, pondo em causa a segurança de todos os comboios envolvidos.

O problema em questão é a criação de alterações nos horários que tenham em conta os desvios, ao horário original, ocorridos e que resultem num plano sem conflitos. Ou seja, numa situação final em que todas as restrições sejam respeitadas. Restrições estas provenientes do problema do *Meet and Pass*, a ausência de colisões frontais e a ultrapassagem realizada em condições controladas de segurança.

A principal ideia desta tese é a criação de um sistema de apoio à decisão que possibilite, numa fase inicial, o auxílio à escolha de alterações a implementar num horário de modo a contornar alguma perturbação, constituindo também uma plataforma para, numa fase posterior, propor automaticamente soluções, sendo que o objectivo máximo é o de um sistema totalmente automatizado que escolha e implemente a melhor solução possível.

A solução foi desenhada para linhas de baixo tráfego, em particular as situações onde apenas existe uma única linha a separar cada par de estações. Este tipo de situações, apesar de longe da complexidade das linhas urbanas e metropolitanas, continua a ser utilizada, principalmente em zonas de grande extensão e onde não se justifica financeiramente a criação de infra-estruturas mais complexas, nem mesmo a utilização de uma linha para cada sentido. Em Portugal existem linhas deste género, por exemplo no Algarve. Contudo, em regiões como a Austrália e o Tibete, estas linhas muitas vezes atingem muitas centenas de quilómetros de comprimento.

A elaboração da solução foi realizada em colaboração com a Alcatel Portugal, actual Alcatel-Lucent, com vista a criar um produto comercial para sistemas de apoio à decisão.

1.1 Descrição dos Capítulos

Um pequeno resumo dos vários capítulos que compõem a dissertação:

Enquadramento

Neste capítulo é feita uma descrição mais detalhada do problema que motivou a realização da presente tese. É feito também um apanhado das soluções existentes ou daquelas que mais se aproximam do problema em questão.

É apresentada a solução proposta e uma introdução à aplicação desenvolvida.

No final, é realizada uma análise comparativa entre as soluções propostas e as existentes, contemplando também a possibilidade de interacção entre as mesmas.

Texto Principal

Neste capítulo é descrito em detalhe a solução proposta. Em particular, é feita uma apresentação do modelo criado para a aplicação e para a gestão de conflitos e uma descrição dos diversos módulos da aplicação desenvolvida. É realizada uma análise das metas definidas no início da tese com as que foram alcançadas no final.

É incluída uma descrição dos vários fluxos de trabalho presentes na aplicação de modo a fornecer uma percepção do funcionamento interno da aplicação.

No final, é feita uma descrição técnica do simulador.

Descrição técnica do simulador

Nesta secção é analisada a aplicação a um nível mais técnico. São apresentadas as razões da escolha da tecnologia utilizada, as opções de projecto tomadas no desenvolvimento da aplicação e uma descrição do modelo utilizado para a aplicação.

Conclusões Finais

Neste capítulo é feita uma introspecção aos resultados esperados face aos obtidos e uma análise à utilidade do resultado final face ao problema a resolver.

Também é feito um levantamento das dificuldades que surgiram durante o desenvolvimento da aplicação, de melhorias que se poderiam introduzir e o trabalho futuro que se poderá realizar no seguimento da presente tese.

Anexo A – Manual de Utilização

O manual da aplicação demonstra a utilização do software e apresenta uma descrição das várias opções presentes na aplicação, das instruções para manipular a informação e os resultados a esperar.

É feita também uma descrição do formato dos ficheiros de entrada e fornecido um exemplo de aplicação.

Anexo B – UML da Aplicação

Os diagramas UML permitem uma percepção da arquitectura da Aplicação criada.

2. Enquadramento

Para um melhor enquadramento e compreensão do texto, torna-se importante descrever a variedade de conceitos específicos do contexto do tráfego ferroviário, sendo que alguns deles foram adaptados com vista a uma simplificação do processo de simulação.

- Comboio – Composição de carruagens movidas por locomotiva;
- Estação de comboio – Local onde um comboio pode parar para demora, embarque, desembarque e mudança de linha (incluindo estações, apeadeiros, pontos de ultrapassagem, etc.);
- Linha – Caminho-de-ferro delimitado por pelo menos duas estações. Uma Linha pode ser constituída por um ou mais Troços;
- Troço – Caminho-de-ferro que liga duas estações. No simulador construído a característica mais importante é a impossibilidade de circulação num Troço de 2 ou mais comboios com sentidos opostos;
- Horário – Tabela que indica as horas a que determinado comboio passa em determinada estação. Consoante o contexto, Horário poderá também ser o conjunto formado pelas horas de chegada e de partida de um comboio para uma determinada estação;
- Cantão – Um troço é composto por diversos cantões. As entradas dos cantões estão sempre sinalizadas com semáforos de modo a que nunca esteja mais do que um comboio em cada cantão. Os cantões acabam por ser na prática uma medida da distância de segurança entre comboios;
- Semáforo – Regulam e delimitam as entradas nos diversos cantões;
- Conflito – Termo utilizado para designar qualquer situação em que um comboio esteja numa situação de desrespeito das regras de segurança obrigatórias ou de perigo eminente.

2.1 – Problema do “*Meet and Pass*” e propagação de conflitos

Na gestão de tráfego ferroviário, ao nível do que é tratado na tese, tipicamente são criados, através de algoritmos próprios, os diversos horários para os vários comboios, de modo a que tudo corra na normalidade e dentro do previsto. Contudo, os horários são elaborados com um buffer de tempo bastante limitado para os atrasos ou seja, sem uma grande margem para contemplar atrasos que possam ocorrer. Este factor possibilita uma maior rentabilidade da linha e dos serviços oferecidos, mas coloca um problema fácil de entender, a simples ocorrência de atrasos, que podem ter uma origem muito variada, irá, com uma grande probabilidade, desencadear uma sequência de novos atrasos, inclusive em outros comboios que partilhem as mesmas linhas e estações.

Designemos um atraso que não deriva de outro como um atraso primário e um atraso que é consequência de outro como um atraso secundário. Um atraso, qualquer um dos tipos, não mais é do que uma mudança no horário do comboio. No entanto é uma mudança forçada pelas circunstâncias, sendo imprevisível no caso dos atrasos primários. Numa outra perspectiva nas alterações nos horários, estão as alterações introduzidas por um gestor de tráfego, que tipicamente são atrasos (em particular para os comboios de passageiros) e que existem simplesmente para corrigir problemas originados por outros atrasos. No entanto, qualquer alteração nos horários gerados inicialmente, seja um atraso ou uma correcção, têm o mesmo problema, podem por si só originar novos conflitos.

Tal como referido no capítulo 1, o problema acaba por se resumir a um problema de *meet and pass* ou seja, a criação de um conjunto de restrições que assegurem o cumprimento das regras de circulação dos comboios e cuja solução final terá sempre de respeitar.

Um conflito ocorre sempre que um comboio é impedido de realizar o seu percurso, tanto espacialmente como temporalmente, no âmbito desta tese foram definidos essencialmente dois tipos de conflitos:

- A entrada de um comboio numa linha ocupada por um ou mais comboios a viajar em sentido oposto.
- A entrada de um comboio num cantão ocupado por outro que viaje na mesma direcção.

Um outro tipo de conflito a detectar, com especial importância quando se estiver a desenvolver uma solução para os conflitos, é a entrada de um comboio numa estação com a capacidade máxima preenchida.

A solução do conjunto destes problemas consiste obrigatoriamente num sistema que sugira alterações nos horários originais, respeitando todas as restrições a cumprir principalmente, mas que simultaneamente tenha a capacidade de prever as consequências das alterações a introduzir. Este facto faz com que haja a necessidade de definir um custo para cada solução, custo esse que serviria de critério de comparação entre várias soluções propostas. Este custo teria que estar necessariamente dependente de uma variedade de factores: o tipo de comboio, os comboios de passageiros têm de cumprir um horário bastante mais rigoroso que por exemplo os comboios de mercadorias; o número correcções a introduzir, provavelmente um atraso grande num horário de um

comboio é mais fácil de implementar do que muitas pequenas alterações a muitos comboios; e outros factores que possam ser relevantes, tais como as estações onde são colocados os atrasos ou a hora do dia em que são realizados.

2.2 – Soluções existentes

O problema da gestão de conflitos é feito actualmente, em particular em Portugal, através de um ou mais operadores que, ao serem confrontados com uma situação que possa gerar um conflito, procuram nas alterações mais comuns uma que corrija o problema e esperam pelo conflito seguinte. Esta abordagem tem a vantagem de utilizar a experiência pessoal dos operadores com um sucesso moderado. No entanto, a falta de capacidade humana para prever situações de uma maior complexidade faz com que as soluções escolhidas venham, muito provavelmente, a originar novos conflitos, descurando inclusive outras soluções que apesar de à primeira vista não parecerem superiores, a médio ou longo prazo poderão originar resultados mais satisfatórios.

As soluções propriamente ditas são compostas tipicamente pela introdução de atrasos em um ou mais comboios, podendo também no caso de transporte de mercadorias haver, embora menos utilizada, a possibilidade de adiantar um horário. Estas escolhas devem-se essencialmente ao facto de nos comboios de transporte de passageiros, existirem garantias aos clientes que se estiverem a determinadas horas na estação de comboio pretendida, irão apanhar o comboio desejado, mesmo que este chegue atrasado. Também existe um outro tipo de soluções, baseada na alteração do percurso do comboio de modo a evitar determinados conflitos.

Na literatura científica é possível encontrar vários trabalhos envolvendo gestão de conflitos. Tipicamente existem dois tipos trabalhos, os baseados em soluções *off-line* ou *clean-slate* e em soluções *on-line* ou incrementais [1,2], as primeiras são aquelas que não são realizadas em tempo real e produzem novos horários e percursos. Nesta categoria estão essencialmente os trabalhos baseados em algoritmos de geração de horários, utilizados na criação inicial de um horário, livre de conflitos, para cada comboio. No segundo tipo de soluções estão todas aquelas cujo objectivo é serem utilizadas por um gestor de tráfego, na altura em que é necessário corrigir variações imprevistas nos horários e cuja solução resida apenas em alterações ao horário original, ao invés da criação de um novo. Este tipo de soluções, face às primeiras, tem uma diferença ainda mais significativa que é a existência de restrições temporais derivadas da necessidade de tomar decisões num curto espaço de tempo.

É nas soluções do tipo *on-line* ou incrementais que se encontra a resposta para o problema tratado nesta tese. Neste campo existem alguns trabalhos relevantes. Pranzo, D’Ariano e Pacciarelli em [2] sugerem uma solução baseada nos problemas de *Job-Shop* mas aplicados ao tráfego ferroviário, onde cada Job corresponde a um comboio e cada bloco corresponde a um cantão, onde apenas pode estar um comboio de cada vez, resultando assim num enorme problema de *Job-Shop*, com centenas de *jobs* e blocos e que na prática corresponde a uma representação com grafos. As soluções são obtidas utilizando diversas heurísticas, originando soluções fortemente baseadas em

rerouting, ou seja, sugestão de percursos alternativos. Também de D'Ariano, [3], surge uma solução baseada em algoritmos de reordenamento, *rerouting* utilizando um modelo de velocidade fixa, com o objectivo de recuperar de perturbações nos horários, em particular atrasos e bloqueios de linha. Esta solução é também baseada na utilização de grafos para representar a linha, utilizando o método *Branch-and-Bound* para chegar à solução final.

Higgins, Ferreira e Kozan em [4] apresentam um modelo para tráfego em linhas únicas, *Single Lines*. Neste trabalho apresentam não só um modelo para optimização de horários mas também um algoritmo para calcular os locais óptimos para a colocação de pontos de cruzamento. A optimização é baseada num sistema de determinação das prioridades de cada comboio envolvido num conflito, decidindo qual é que fica à espera no ponto de cruzamento ou paragem mais próximo. Esta solução utiliza também o método *Branch-and-Bound* para chegar à solução óptima.

Ahuja em [1] propõe um sistema mais abrangente, também baseado em grafos, possibilitando tanto soluções off-line como on-line, tendo em consideração o fluxo de carruagens, de locomotivas e de tripulação. Este sistema apoia-se fortemente em *rerouting* e *reordering*, procurando caminhos que levem a percursos mais rápidos.

Missikoff, [5], apresenta uma abordagem orientada a objectos, a incidir fortemente na componente de monitorização do estado das linhas e dos comboios em tempo real. Na detecção ou previsão de um conflito adopta duas possíveis soluções, colocar um dos comboios a andar mais lentamente ou fazer com que um deles pare num ponto de paragem para que se cruzem ou se faça uma ultrapassagem.

Uma última referência ao projecto LOCROPOL, [6], um sistema a ser desenvolvido para a utilização em linhas de baixo tráfego e que utiliza satélites juntamente com sensores nos comboios e nas suas rodas, possibilitando saber em tempo real onde se encontram os comboios e fazer uma gestão do tráfego mais segura. Actualmente, durante a circulação dos comboios não se sabe onde estes se encontram. Existem sensores espalhados nas linhas e que indicam quando é que o comboio lá passou.

2.3 – Solução Proposta

A solução proposta surgiu de uma necessidade específica no mercado português e da empresa Alcatel-Lucent, gestão de tráfego ferroviário em linhas de baixo tráfego, consistindo numa série de etapas a desenvolver.

O ideia geral é a criação de uma aplicação que, em tempo real, ou seja, com o objectivo de ser utilizada directamente por um operador, permita resolver directamente conflitos que venham a surgir. Pretende-se, a partir do horário gerado inicialmente ou de um que origine já pelo menos um conflito, possibilitar a introdução de alterações, prever as consequências dessas mesmas alterações, calcular várias soluções que resolvam os conflitos detectados e, baseando-se numa função de custo, escolher a melhor solução. Inicialmente o sistema seria isolado, mas numa fase posterior poderia vir a receber automaticamente os horários reais, incluindo as variações provocadas pelos conflitos até à data. O

sistema é baseado num modelo de velocidade fixa, não considerando acelerações nem desacelerações, utilizando apenas soluções baseadas em alterações nos horários dos comboios e não considerando a gestão do tráfego local em cada estação.

O trabalho da presente tese corresponde ao desenvolvimento do sistema de previsão de conflitos, em particular, um simulador de eventos discretos que partindo de um determinado horário, com ou sem alterações introduzidas pelo operador, possa detectar o sítio e o local de futuros conflitos. Esta fase da aplicação não resolve o problema da gestão de conflitos. Fornece uma ferramenta que possibilite uma gestão, por parte de um operador, mais simples, rápida e eficaz, produzindo uma informação mais rica sobre as consequências das alterações introduzidas.

O simulador recebe uma descrição da linha, a descrição dos comboios e os horários correspondentes e produz uma descrição escrita ou gráfica dos conflitos detectados. A acompanhar, existe uma interface gráfica que permite ao utilizador manipular os horários introduzidos e observar as consequências.

2.4 – Análise comparativa da solução proposta com as soluções existentes

A solução proposta não adopta uma estratégia de mudança de percurso, preferindo limitar-se a produzir mudanças nos horários.

A componente desenvolvida nesta tese, o simulador de tráfego, não resolve por si só o problema da gestão de conflitos. Em conjunto com o que se faz actualmente, ou seja, operadores a introduzir alterações “típicas”, origina uma melhoria dos serviços. Com as duas soluções combinadas, à potencialidade do simulador em detectar conflitos difíceis de prever junta-se a experiência de um operador na resolução de conflitos.

A solução proposta, face aos algoritmos de reordenamento e rerouting, tem a vantagem de não mudar intrinsecamente a metodologia actual, reduzindo assim os custos de implementação. No entanto, o simulador actua sobre um horário que lhe é introduzido e sendo assim, a utilização de um algoritmo que inclua reordenamento ou mudanças de rota, além de mudanças nos horários, é-lhe transparente. Este facto possibilita que seja possível servir de complemento a algumas das soluções existentes.

3. Texto Principal

Como foi enunciado na secção 2.3, a solução proposta não é por si só resposta para o problema da gestão de conflitos de tráfego ferroviário, mas sim uma ferramenta que origina uma evolução e um aumento de eficácia à solução praticada nos dias de hoje, em particular em Portugal.

Ainda de realçar que o tipo de Linhas de comboio para a qual a aplicação foi originalmente desenvolvida são as Linhas de baixo tráfego, ou seja, linhas com uma utilização, em termos de comboios e viagens, bastante inferior à de por exemplo uma linha metropolitana e com apenas uma linha para os dois sentidos.

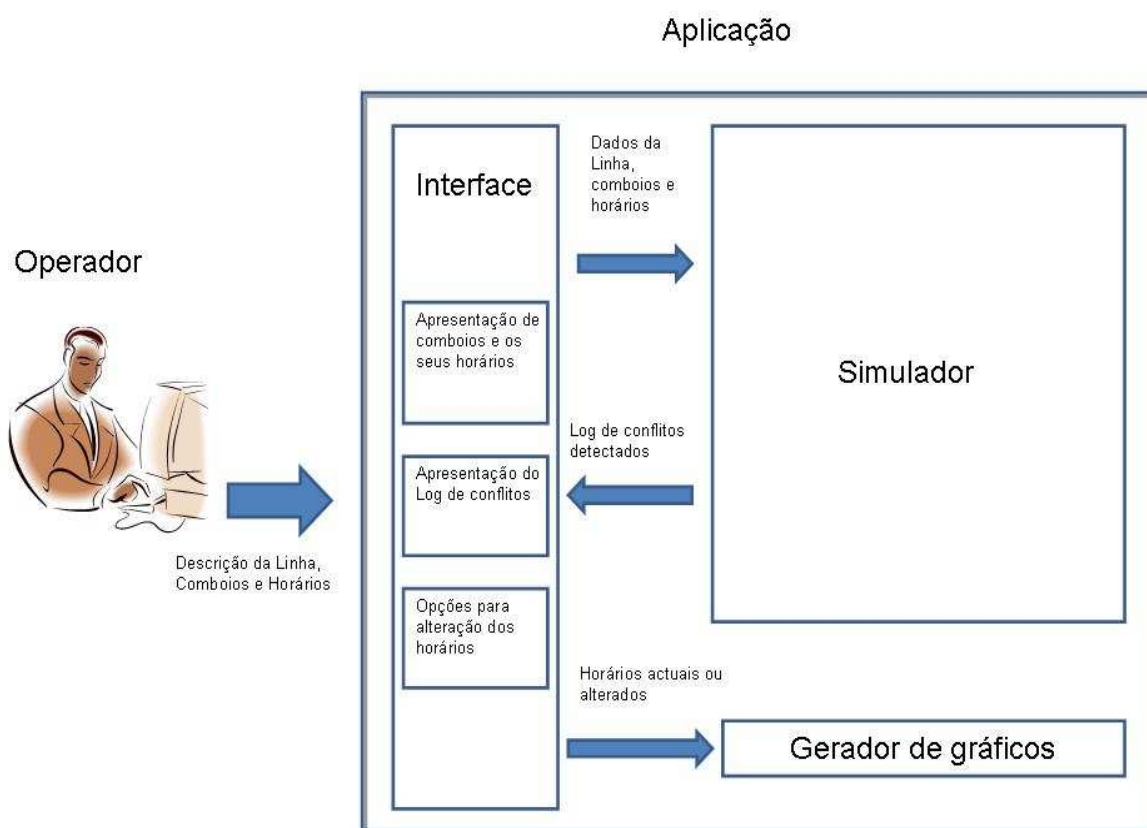


Figura 1 - Estrutura geral da aplicação

O produto final é uma aplicação, desenvolvida na linguagem de programação JAVA [7,8,9], que como se pode ver na Figura 1, é composta por:

- Uma interface, que possibilita que o operador introduza os dados necessários para a simulação, que altere esses mesmos dados de entrada (apenas horários) de

modo a, não só introduzir conflitos, mas também a introduzir alterações que possam resolver esses mesmos conflitos.

- O simulador de eventos discretos, recebe uma descrição das Linhas, dos Comboios, e dos seus Horários e que com essa informação, simula todo o andamento dos comboios, ao mesmo tempo que regista num Log todos os conflitos que vai detectando.
- Um gerador de gráficos, que consiste numa ferramenta para ajudar na interpretação do Log. O Log, depois de recebido é apresentado na Interface através de uma lista dos conflitos previstos. O operador tem a possibilidade de chamar o gerador de gráficos para ficar com uma representação mais global do andamento do resultado da simulação. Antes de fazer a geração do gráfico é possível efectuar alguma customização.

Como será mostrado na secção 3.2, existem várias maneiras de utilizar a aplicação desenvolvida. Para uma melhor compreensão da sua utilidade, segue um exemplo da resolução de um conflito simples (Figura 2).

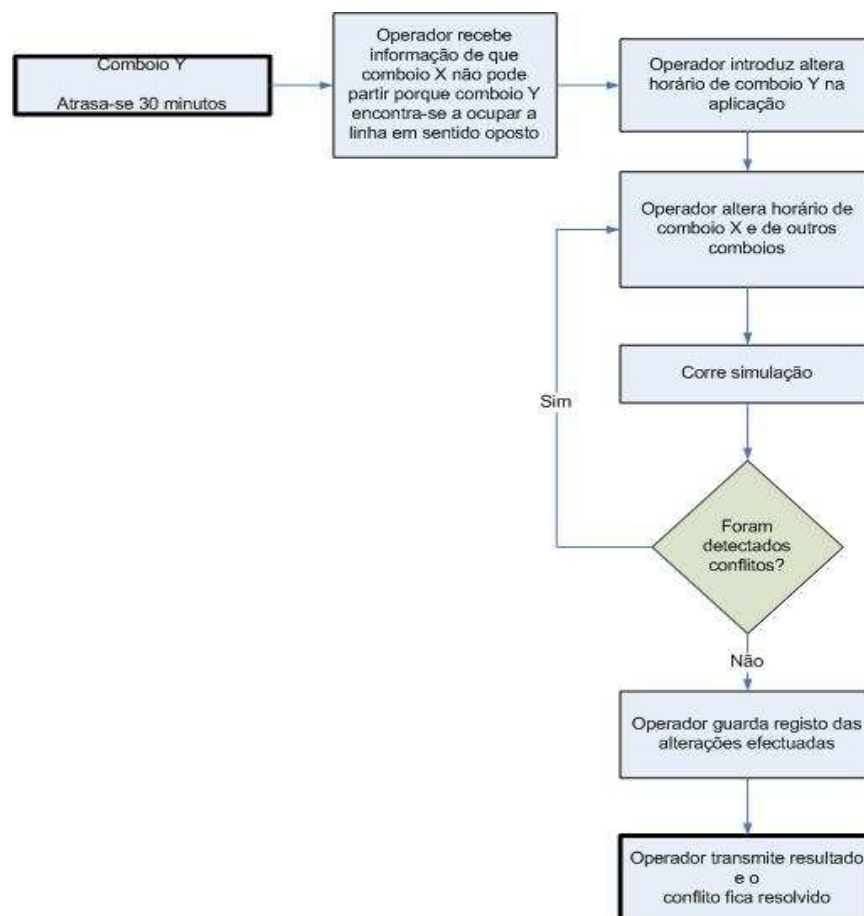


Figura 2 - Exemplo de utilização da aplicação

3.1 Opções de projecto (nível global)

Opções de projecto tomadas a nível conceptual, em particular para o simulador e a interface com o utilizador.

Simulador

A planificação inicial do projecto incidiu especialmente na componente do simulador. A ideia original consistiu em desenvolver uma aplicação que, com base na especificação de uma linha, um ou mais comboios e os respectivos horários, simulasse simultaneamente o horário de cada um e detectasse potenciais conflitos.

O modelo escolhido para a Linha pode ser visualizado na Figura 3. Uma estação é qualquer sítio onde existem pelo menos uma linha dupla e onde um comboio possa parar e/ou mudar de linha, o que além das estações no sentido convencional, inclui também por exemplo os pontos de cruzamento. Um Troço é o caminho que liga duas estações e é composto por pelo menos um cantão.

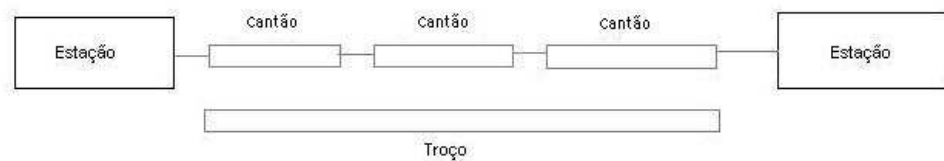


Figura 3 – Modelo da linha de comboio

Uma importante consideração a tomar em conta é o facto de um Troço ter apenas uma linha, o que faz com que cada Troço, desde que tenha um comboio a circular, tenha apenas um sentido definido. Se se pretendesse modular um sistema em que haja duas linhas a separar duas estações, estas posteriormente seriam tratadas como dois troços independentes.

O modelo foi projectado para as linhas de baixo tráfego, em que duas estações estão ligadas apenas por uma linha para os dois sentidos e que cada estação esteja apenas ligada a duas Linhas. Contudo, como será explicado mais à frente na secção 3.3, a tecnologia de programação por objectos utilizada permite facilmente dotar o simulador da capacidade de, tendo o destino do comboio e a indicação da Linha que irá seguir, permitir que haja entre duas estações várias linhas de ligação e que cada estação esteja ligada a mais do que duas linhas.

De qualquer modo, e em particular em Portugal, só recentemente é que os ficheiros de horários produzidos começam a aparecer com o campo da linha destino do comboio preenchido. Na maior parte dos casos continua a ser decidido localmente, na estação, para que linha é que o comboio seguirá.

O próprio modelo da Linha indica já o tipo de conflitos considerados:

- *Dois comboios em sentido contrário no mesmo Troço*, correspondente ao problema de *Meet* - dado que os troços são uma linha única e não têm zonas onde possam mudar de linha, um conflito deste género resultaria obrigatoriamente em acidente. A detecção deste tipo de conflito, ao nível do simulador, resulta do facto de quando um comboio entra num troço, a esse mesmo troço é atribuída uma direcção, quando um outro comboio a viajar numa direcção oposta tenta entrar no troço, é imediatamente detectado o conflito.
- *Dois comboios no mesmo cantão*, correspondente ao problema de *Pass* - apesar de ser um conflito pouco frequente, já que apenas em troços de dimensões consideráveis são colocados dois comboios no mesmo troço, não deixa de ser uma possibilidade. Um cantão, por definição, consiste num espaço na linha onde apenas pode estar um comboio de cada vez, constituindo assim uma medida de segurança para a distância entre comboios. Os cantões podem ter tamanhos variáveis dentro do mesmo troço e são delimitados por semáforos. Ao nível do simulador, a sua detecção será descrita em detalhe mais à frente.
- *Estação sobrelotada* – Este tipo de conflito é o mais raro de acontecer, mas que tem necessariamente que ser contabilizado. Na simulação cada estação foi projectada para ter um número máximo de comboios, especificada pelo utilizador, e um contador dos mesmos. Quando um comboio entra numa estação, o contador é actualizado e comparado com o valor máximo.

O tratamento de eventos foi o aspecto que mais mudanças sofreu desde a ideia inicial até ao resultado final.

Na versão final, um evento é criado sempre que um comboio entra ou sai de uma estação. Ao entrar, é criado um evento com a hora de partida e, ao tratar o evento de partida, é criado um evento para a hora de chegada à estação seguinte. Inicialmente, a lista de eventos do simulador é preenchida com um evento da primeira partida de cada comboio. Este tratamento de eventos implica que, quando um comboio entra num troço ocupado por outro comboio no mesmo sentido, não se saiba à partida o cantão onde se encontra o comboio que está nesse momento no troço.

No entanto, como uma das informações fornecidas na descrição da linha é a velocidade média em cada cantão, conjugando com o tempo de viagem que é conhecido, pode-se extrapolar o tempo que cada comboio está em cada cantão. Realizando cálculos semelhantes para o último comboio que tenha entrado no troço, passa a ser possível detectar quando um comboio entra num cantão ocupado.

Uma opção de destaque que teve várias interpretações foi o critério de paragem da simulação. A ideia inicial consistia em parar a simulação logo que se encontrasse o primeiro conflito. No entanto, utilizando esta abordagem poder-se-ia estar a perder a informação de alguns conflitos posteriores que seriam independentes do primeiro conflito detectado, perdendo-se assim uma visão global dos problemas.

Na versão final, a simulação decorre até não existirem mais eventos, assinalando num Log todos os conflitos que sejam detectados. Esta abordagem tem a desvantagem de registar uma série de conflitos que, como são posteriores ao primeiro conflito que iniciou o Log, têm a possibilidade de nunca virem a acontecer. De modo a conduzir a uma melhor interpretação do ficheiro de Log, todos os conflitos detectados, envolvendo um comboio que anteriormente já tenha originado um conflito, são marcados com um “*” no início do registo.

Na versão inicial, um evento era criado sempre que um comboio entrasse ou saísse de uma estação ou de um cantão. O tempo de viagem em cada cantão era calculado através das equações do movimento mais simples (equação (1) e (2)). O tempo de viagem calculado era composto pelo tempo de aceleração (à aceleração máxima do comboio) até à velocidade média do Cantão, o tempo de desaceleração até atingir a velocidade média do cantão seguinte (no caso de a seguir vir uma estação, a velocidade seguinte era de 0 Km/h) e pelo tempo à velocidade média do segmento.

$$x = x_0 + vt + \frac{1}{2}at^2 \quad (1)$$

$$v = v_0 + at \quad (2)$$

As principais vantagens desta aplicação são as seguintes:

- Simplificação da detecção de dois comboios no mesmo cantão pois bastava verificar se o cantão seguinte estava ocupado.
- Uma maior precisão no tempo de viagem em cada cantão devido ao cálculo dos tempos de aceleração e desaceleração.

No entanto, contém algumas desvantagens:

- A mais importante e incontornável é o facto de, na prática, ser muito difícil obter os dados necessários de cada comboio; velocidade máxima, aceleração e desaceleração máxima. Não havendo a possibilidade de obter estes dados, apesar de teoricamente a solução funcionar, deixava no entanto de ser praticável.

- Para cada viagem, de cada comboio, existiria sempre um número de eventos proporcional ao número de cantões, ao invés de um evento por troço.

Apesar de tudo, o facto de apenas tardiamente se conhecer a impossibilidade de obter toda a informação dos comboios e o facto de o modelo já estar construído e a funcionar, optou-se por alterar o funcionamento da gestão dos eventos para o modelo final, mas deixar no código toda a estrutura necessária para, num seguimento futuro desta tese, se poder utilizar as equações do movimento.

Interface

A interface da aplicação permite uma utilização mais intuitiva e clara da ferramenta de simulação. Foram definidos os seguintes módulos para a sua constituição:

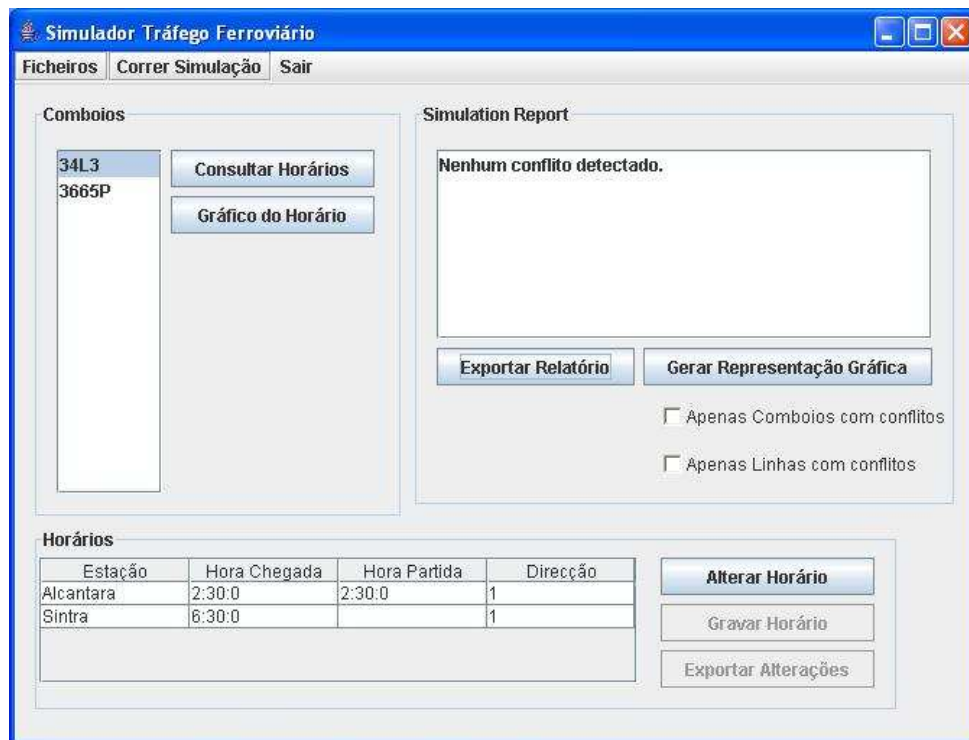


Figura 4 – Ecrã principal da aplicação

- Menu – No menu é possível carregar dados de entrada, correr a simulação e sair da aplicação. O carregamento de ficheiros tem de seguir uma ordem pré-definida e para tal as opções de carregamento, ou para correr a simulação, vão sendo disponibilizadas à medida que os carregamentos vão sendo efectuados. A ordem é a seguinte: Estações->Comboios->Linha ->Horários->Correr Simulação.

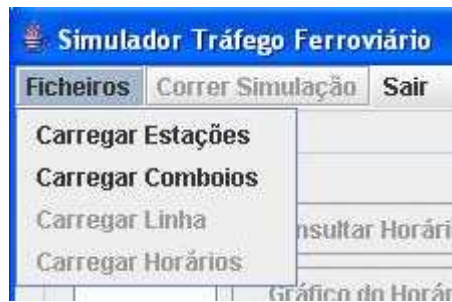


Figura 5 – Detalhe das opções no menu da aplicação

- Comboios – Neste módulo é apresentada uma lista dos comboios existentes. É também possível pedir o horário do comboio seleccionado e a sua representação gráfica.

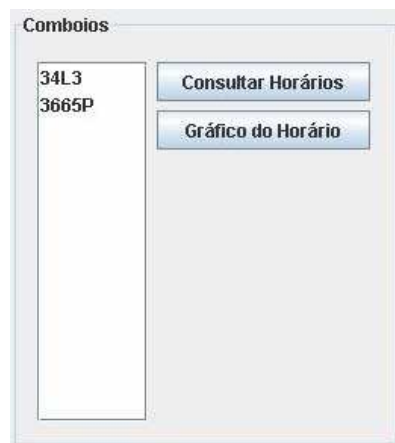


Figura 6 – Detalhe da secção de comboios

- Horários – Apresentação do Horário de um comboio previamente escolhido. Para cada item do horário é apresentado o nome da estação, hora de chegada, hora de partida e direcção. Ao tratar-se da partida inicial do comboio, a hora de chegada é igual à hora de partida. Quando se trata de uma passagem sem paragem, hora de chegada fica a vazio e ao tratar-se da chegada à última estação, a hora de partida fica a vazio. A direcção é definida como 1 ou -1 consoante o sentido.

É também neste módulo que se introduz as alterações nos horários. Estas alterações podem representar os conflitos a introduzir ou as soluções a testar. Também está contemplada a possibilidade de gravar o horário alterado.

Horários

Estação	Hora Chegada	Hora Partida	Direcção
Alcantara	2:30:0	2:30:0	1
Sintra	6:30:0		1

Figura 7 – Detalhe da secção de Horários

- Simulation Report – Depois de efectuada a simulação é apresentado o Log dos conflitos detectados. Neste módulo é também possível pedir uma representação gráfica dos horários dos comboios, através do gerador de gráficos. Antes de a pedir é possível definir alguns parâmetros, tais como: intervalo temporal, inclusão de apenas as linhas que envolvam conflitos ou ainda apenas os horários dos comboios envolvidos nos conflitos.

Simulation Report

Nenhum conflito detectado.

Apenas Comboios com conflitos

Figura 8 – Detalhe da secção de relatório de simulação

- Gerador de gráficos – Os gráficos gerados permitem uma representação espaço-tempo dos horários dos vários comboios. Os dados de entrada para gerar os gráficos são os presentes horários dos comboios (depois de sofrerem alterações ou não).

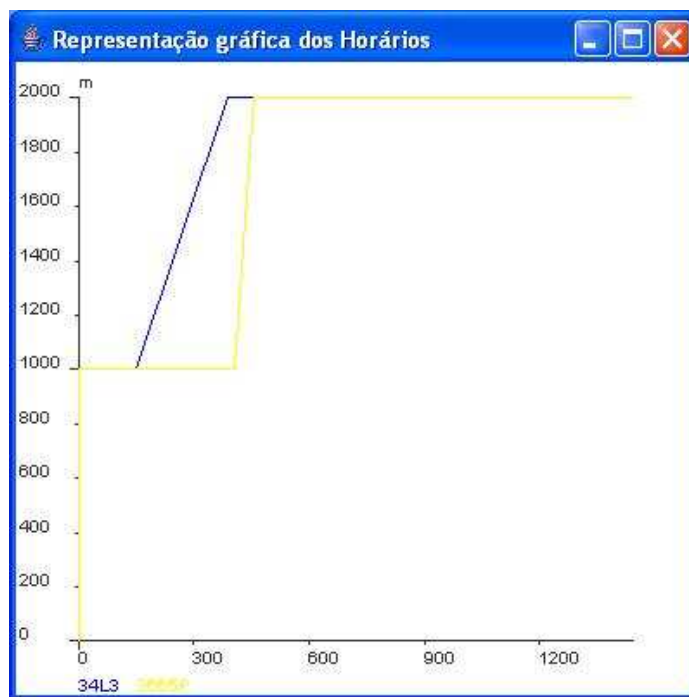


Figura 9 – Exemplo de gráfico gerado

Inicialmente o principal elemento de análise era a representação gráfica. Contudo, após alguma deliberação decidiu-se dar mais ênfase à representação textual (que acaba por ser bastante mais precisa) e deixar a representação gráfica apenas para permitir uma visão mais global da posição relativa dos comboios. No eixo das abcissas encontra-se o tempo em segundos e no eixo das ordenadas a posição do comboio em metros.

Síntese

No panorama geral o projecto da aplicação cumpre os requisitos para as quais foi inicialmente concebido.

Como qualidade adicional está o facto de a estrutura do simulador permitir tratar mais do que uma linha em simultâneo, ou seja, as estações existentes não têm que estar todas ligadas formando um único caminho-de-ferro.

Um ponto que não resultou como o que estava previsto inicialmente foi a representação gráfica que acabou por ficar menos rica do que estava esperado inicialmente. Talvez a maior desvantagem da tecnologia utilizada foi a não existência de uma função plot de raiz. A criação de uma função do género é material suficiente para um trabalho de fim de curso só por si e, sendo assim, foi utilizado um gerador de gráficos OpenSource em Java.

3.2 Descrição Funcional da Aplicação

Esta secção é composta essencialmente por um conjunto de fluxogramas que proporciona uma visão mais compreensiva do funcionamento da aplicação.

Os processos descritos vão desde os mais gerais (funcionamento geral da aplicação) até aos mais particulares (por exemplo o cálculo do tempo de viagem de um comboio entre duas estações).

Interacção com a aplicação

O processo de interacção com a aplicação vai desde carregar os ficheiros de entrada até ao teste de soluções para a resolução de conflitos.

Algumas notas importantes:

- A alteração de horários tanto pode ser para introduzir conflitos ou para introduzir soluções para os conflitos.
- Os Horários carregados tanto podem ser os iniciais, como horários já alterados e que produzem conflitos.

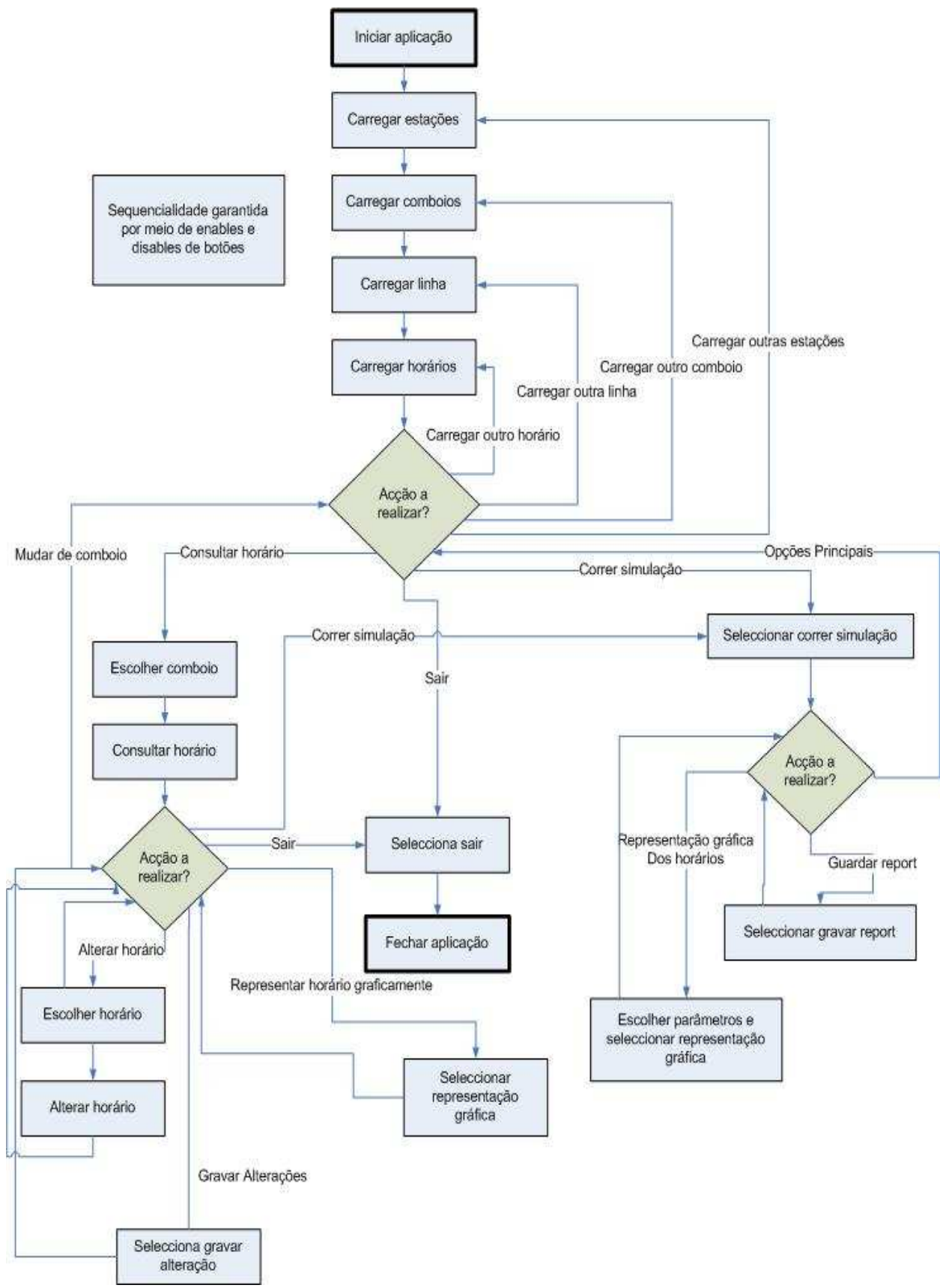


Figura 10 – Processo de interacção com a aplicação

Carregamento de Dados (geral)

O processo de carregamento de dados é idêntico para todos os dados de entrada (Estações, Comboios, Linhas e Horários). As diferenças residem essencialmente na captura dos dados que varia com o formato de cada ficheiro.

O carregamento dos horários produz não só uma lista de horários para cada comboio, mas cria também um evento por comboio.

Os detalhes do carregamento de cada ficheiro de entrada podem ser consultados em anexo.

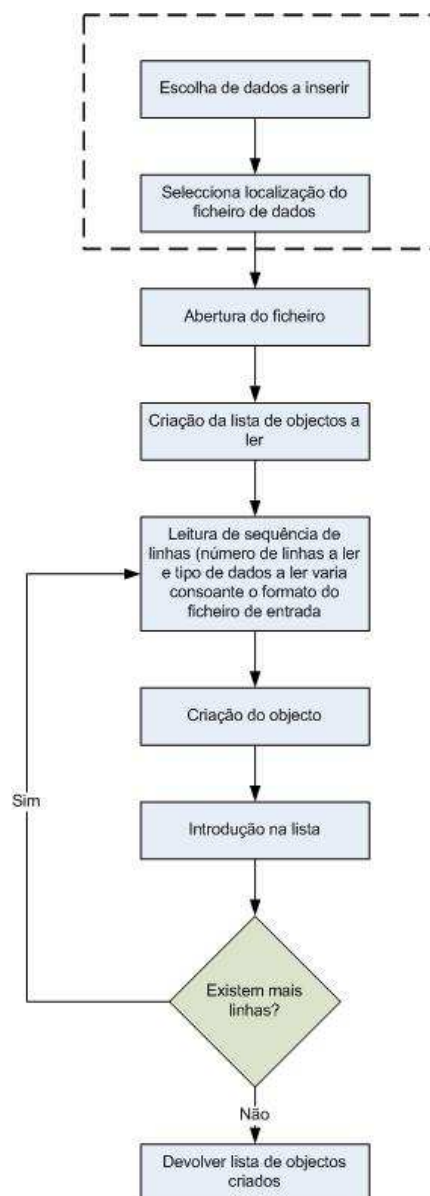


Figura 11 – Processo geral de carregamento de dados

Carregamento de Estações

O processo de carregamento de estações recebe do utilizador uma referência para o ficheiro com as definições. Como resultado final, devolve uma lista das estações presentes no ficheiro.

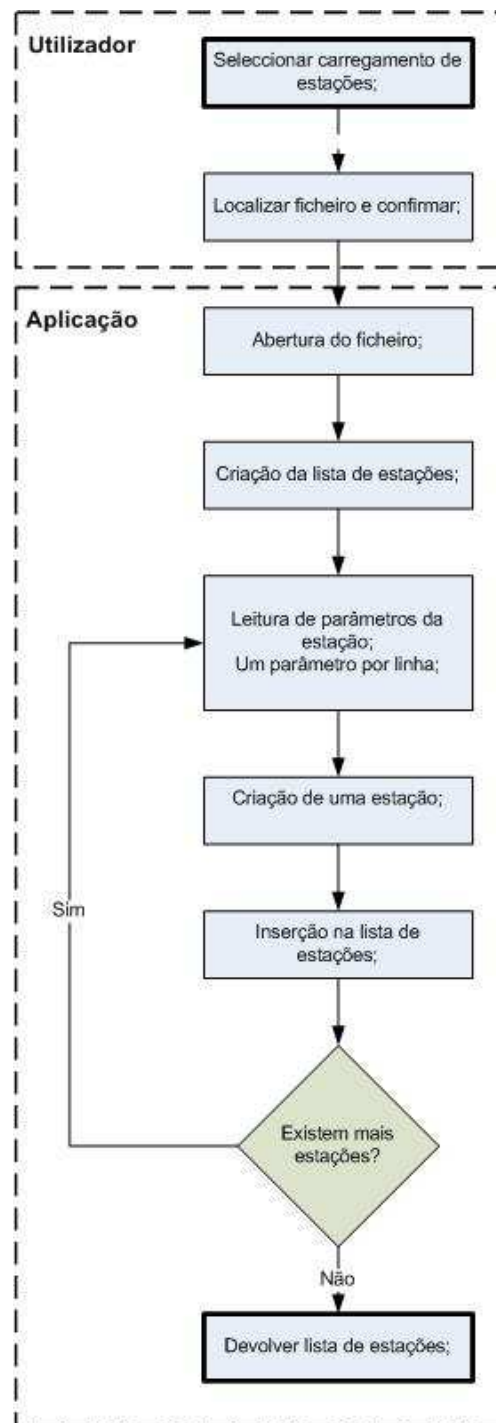


Figura 12 – Processo de carregamento de estações

Carregamento de Comboios

O processo de carregamento de comboios recebe do utilizador uma referência para o ficheiro com as definições. Como resultado final, devolve uma lista de comboios.

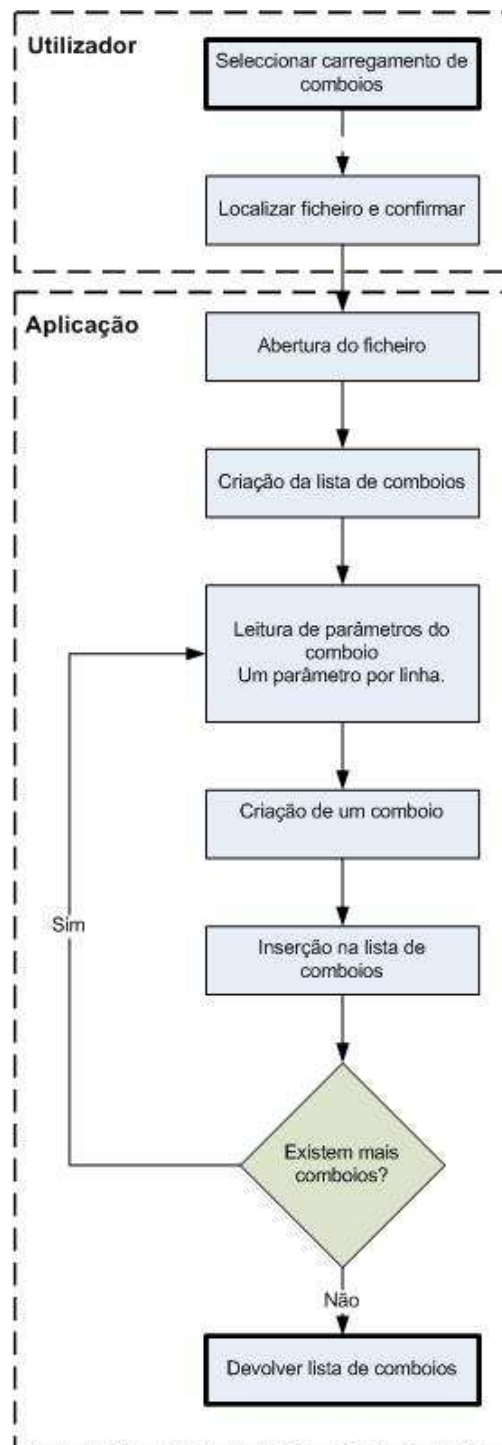


Figura 13 – Processo de carregamento de Comboios

Carregamento das Linhas

O simulador foi projectado para simular os comboios em apenas uma linha. Contudo o seu processo de funcionamento permite-lhe tratar várias linhas simultaneamente,

O processo recebe a referência do ficheiro com as características das linhas e a lista de estações já previamente inseridas. Ao criar os Troços e cantões é feita a ligação entre eles e a ligação com as estações. Desta forma fica sempre possível aceder à estrutura das linhas a partir das várias estações.

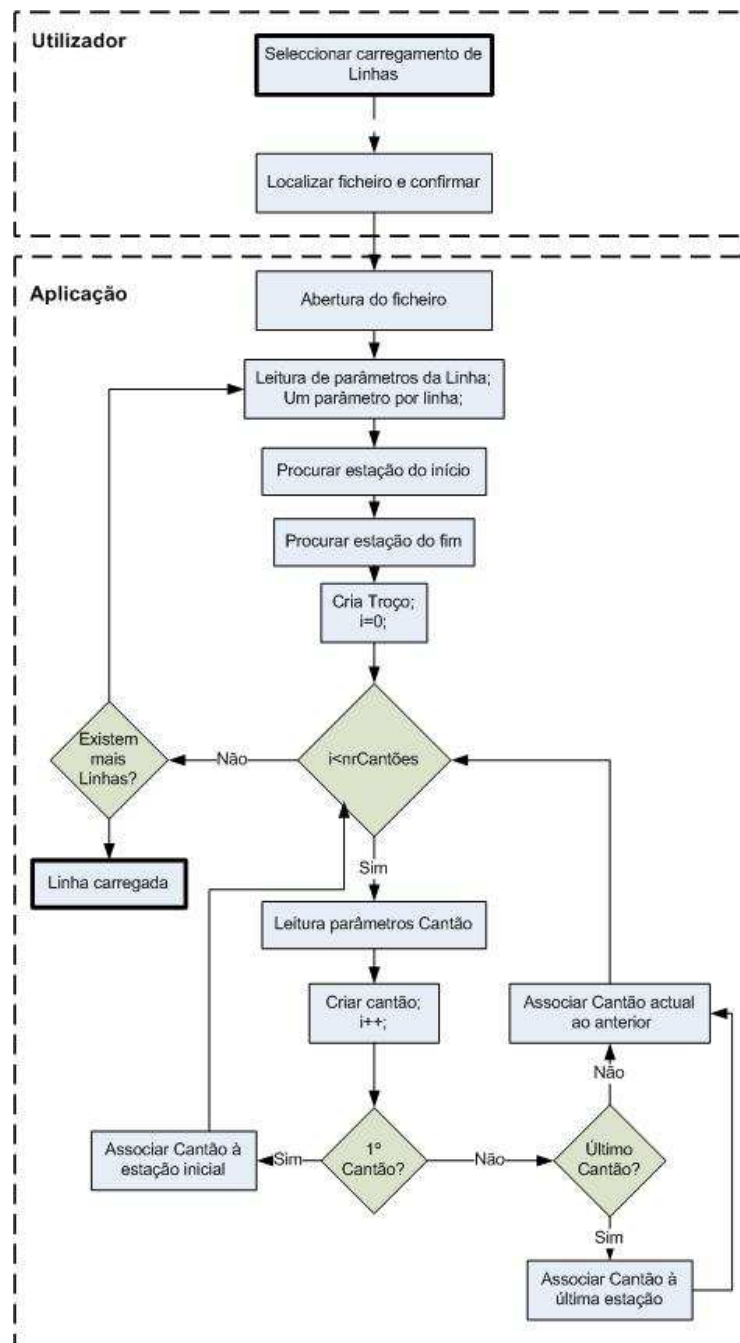


Figura 14 – Processo de carregamento de linhas

Carregamento dos Horários

O processo de carregamento de horários é o mais complexo e exige que já tenha sido criada toda a representação da linha, estações e comboios.

O ficheiro fornecido pelo utilizador contém para cada comboio, os horário de chegada e partida para cada estação. À medida que se vai lendo o ficheiro, vão-se criando os horários de cada comboio. Foram definidos essencialmente dois tipos de representação do horário. Uma baseada no formato do ficheiro original, que contém a hora de chegada e partida para cada estação. Esta representação é utilizada posteriormente para a geração de representação gráfica e para uma melhor identificação das alterações introduzidas nos horários. A segunda representação contém a hora de partida e o tempo de viagem, esta representação é a que é utilizada na simulação do horário.

Para o primeiro horário de cada comboio é também criado um evento de partida. Desta forma, no final da leitura, a lista de eventos conterá um evento para a primeira partida de cada comboio.

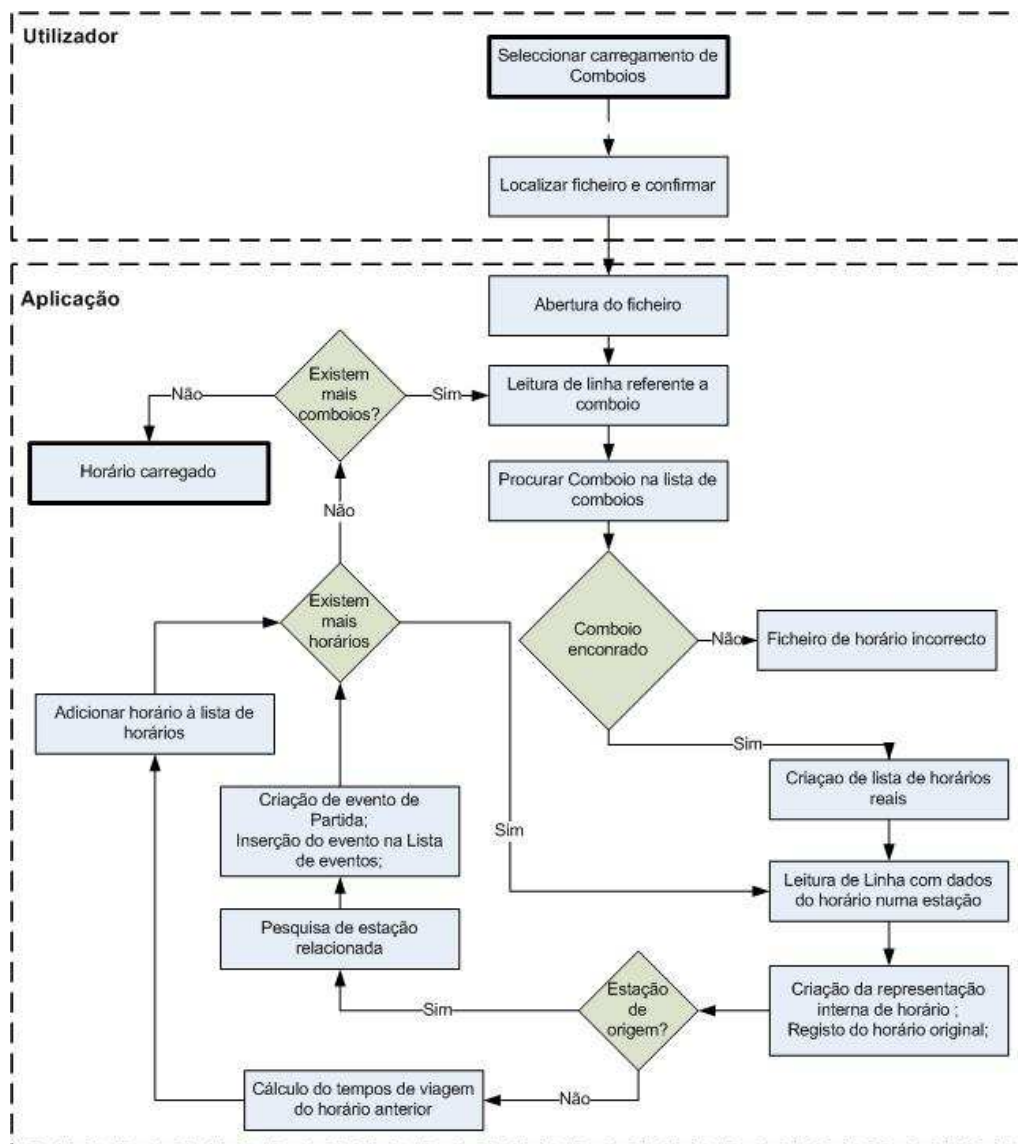


Figura 15 – Processo de carregamento de horários

Gravação de Log

Processo de gravação do relatório dos conflitos detectados e da alteração dos horários. Depois de efectuada uma simulação, o utilizador tem a possibilidade de registar num ficheiro o relatório gerado (Figura 16).

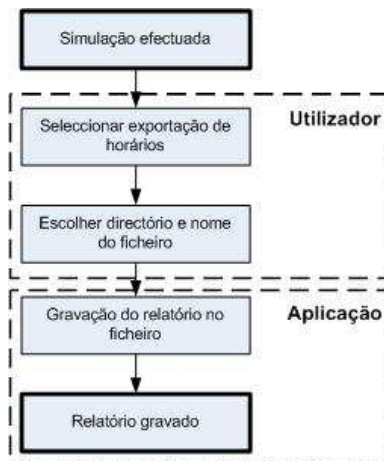


Figura 16 – Processo de gravação de log

Gravação de Horários

A aplicação permite também fazer um registo das alterações efectuadas. Foram desenvolvidos dois formatos diferentes para guardar as alterações. A “exportação das alterações” consiste em guardar num registo apenas as alterações efectuadas. A opção de “guardar as alterações” regista num ficheiro uma representação completa do horário do comboio em questão (Figura 17).

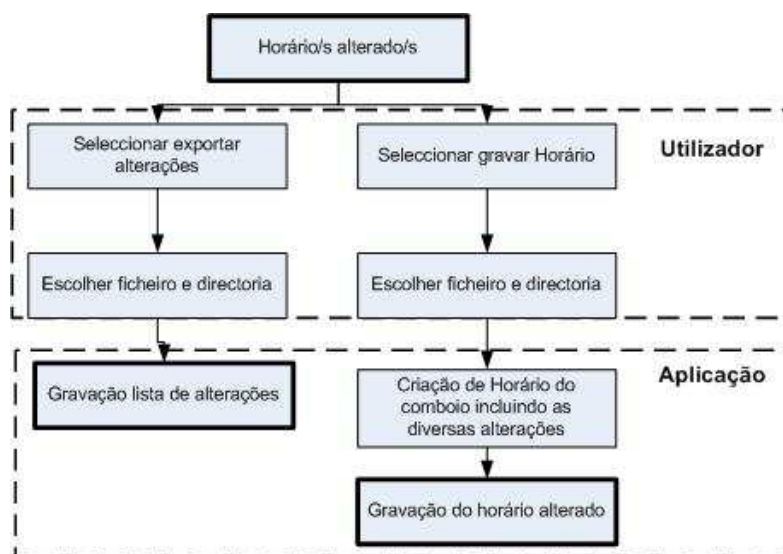


Figura 17 – Processo de gravação das alterações introduzidas nos horários

Geração gráfica dos Horários

A geração de gráficos dos horários dos comboios permite obter uma visão global da distância espaço-temporal entre os vários comboios. A figura 18 corresponde à geração dos gráficos dos comboios depois de efectuada a simulação. O utilizador tem ainda a possibilidade de escolher que sejam gerados apenas os horários dos comboios presentes na linha onde foram detectados conflitos, ou apenas os horários dos comboios que estão envolvidos nos conflitos detectados.

Uma outra possibilidade, descrita na figura 19, é a geração do gráfico do horário de um comboio em particular (Figura 18 e 19).

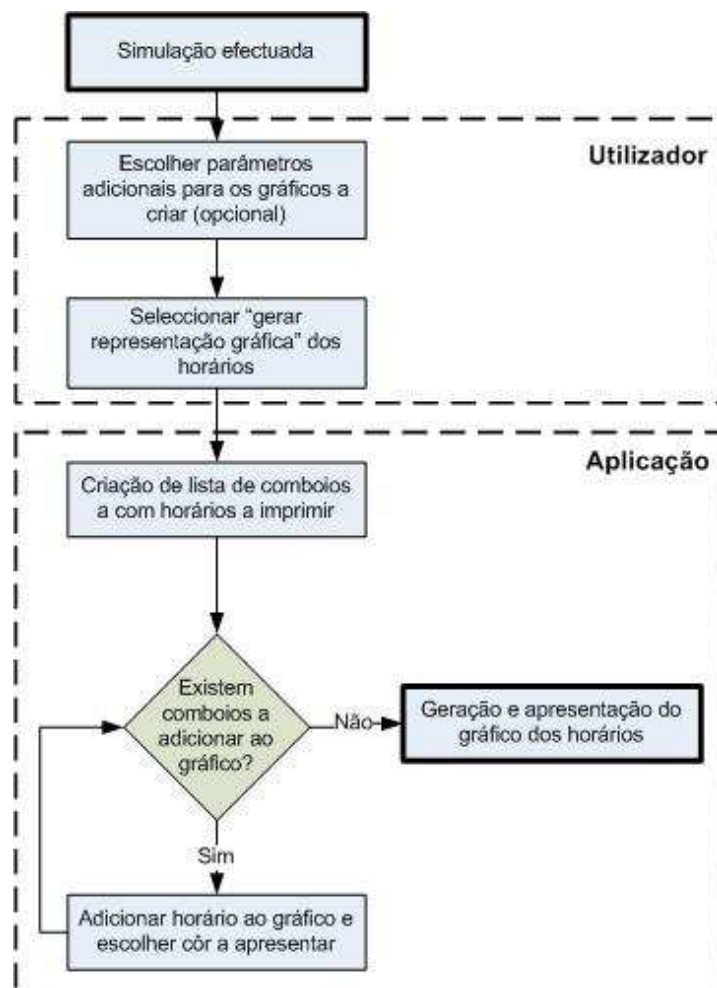


Figura 18 – Processo de criação de um gráfico dos horários dos vários comboios

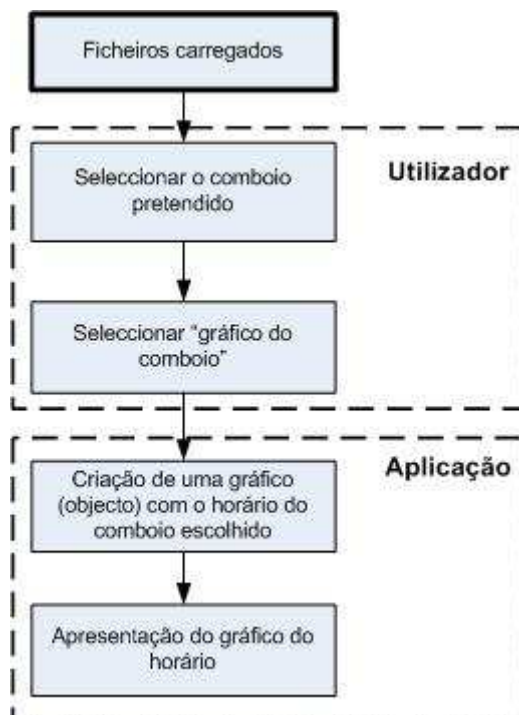


Figura 19 - Processo de criação do gráfico do horários de um comboio

Simulador (geral)

O processo do simulador é o cerne da aplicação. Depois de ter todos os dados carregados, o sistema permite a simulação dos horários introduzidos (Figura 21).

O simulador funciona por evento discretos, sendo que um evento é criado sempre que um comboio sai ou entra numa estação. A qualquer altura da simulação, o número de eventos existentes é sempre igual ao número de comboios que se encontram a circular ou que ainda não o começaram a fazer.

Ao longo de todo o processo, vão sendo registados os conflitos que forem sendo detectados, chegando a simulação ao fim quando não existirem mais eventos por tratar.

Simulador (tratamento de chegadas)

O processo de tratamento de chegadas ocorre sempre que um comboio chega ao seu destino. Caso a estação esteja lotada, é registado um conflito devido à ultrapassagem da capacidade da estação. Havendo ou não detecção de conflito, posteriormente o sistema consulta o horário do comboio para saber se existem mais partidas previstas, caso não haja, o comboio deixa de produzir eventos. Se ainda existirem partidas, é verificado se a partida é imediata (comboio de passagem na estação) ou se falta algum tempo. No primeiro caso, é chamado o processo de tratamento de partida, no segundo, é actualizado o tempo de espera do comboio e criado um evento de partida (Figura 22).

Simulador (tratamento de partidas)

O processo de tratamento de partidas é realizado sempre que um comboio parte de uma estação (Figura 23) .

Antes da partida propriamente dita, é verificado se a partida originará conflitos. Começa por verificar se existe um comboio, no troço para onde vai entrar, que esteja no sentido oposto, caso exista, é registado um conflito. De seguida, caso já exista um comboio no troço com o mesmo sentido, verifica se em alguma altura estarão simultaneamente a ocupar o mesmo cantão. Para fazer esta verificação, o sistema recorre à distância de cada cantão e à velocidade média em cada um deles (Figura 20).

No final do processo é criado um evento de chegada para o comboio.

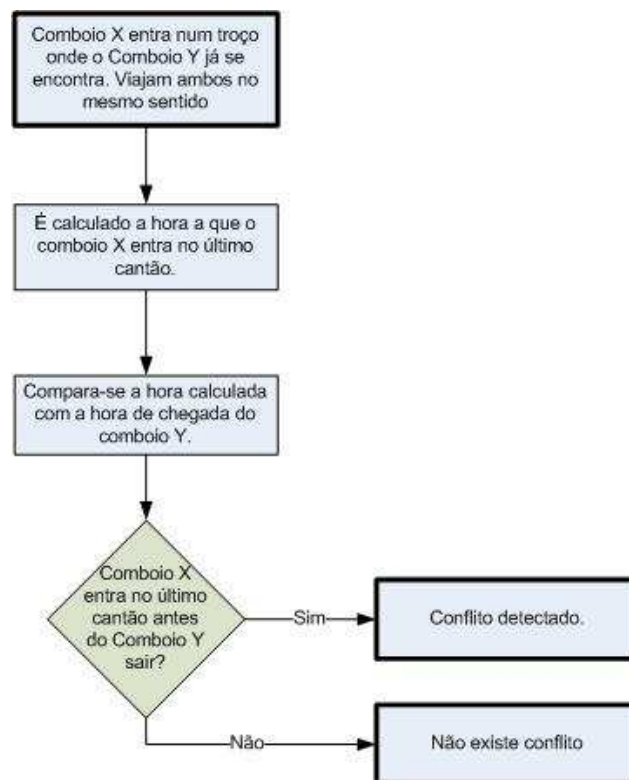


Figura 20 Detecção de conflitos do tipo Pass

O cálculo da hora de entrada no último cantão do comboio X (ver Figura 20) é feito segundo a equação (3).

$$\sum \frac{distância_cantão(i)}{velocidade_média_cantão(i)} + Hora_entrada_no_troço \quad (3)$$

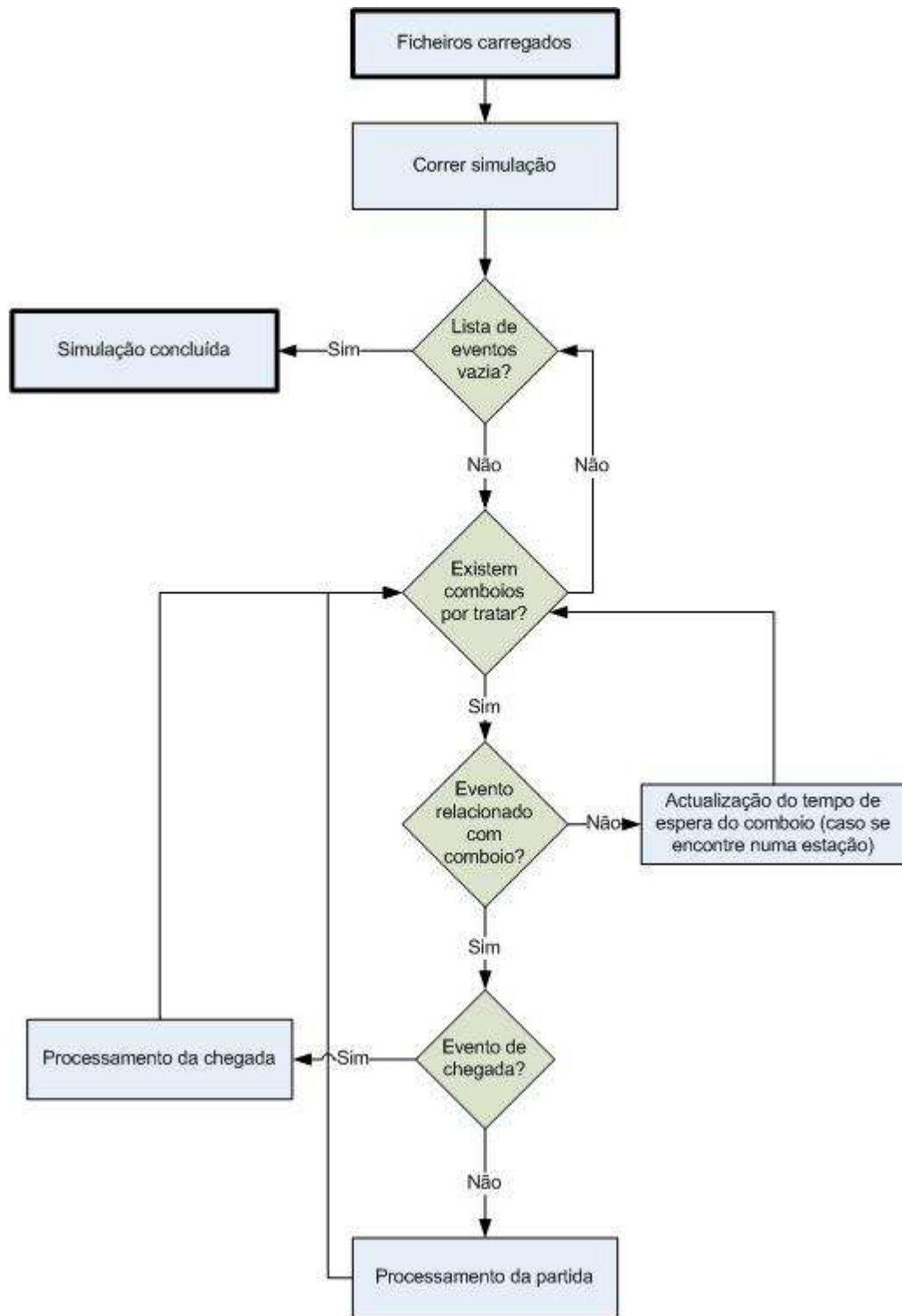


Figura 21 – Processo geral do funcionamento do simulador

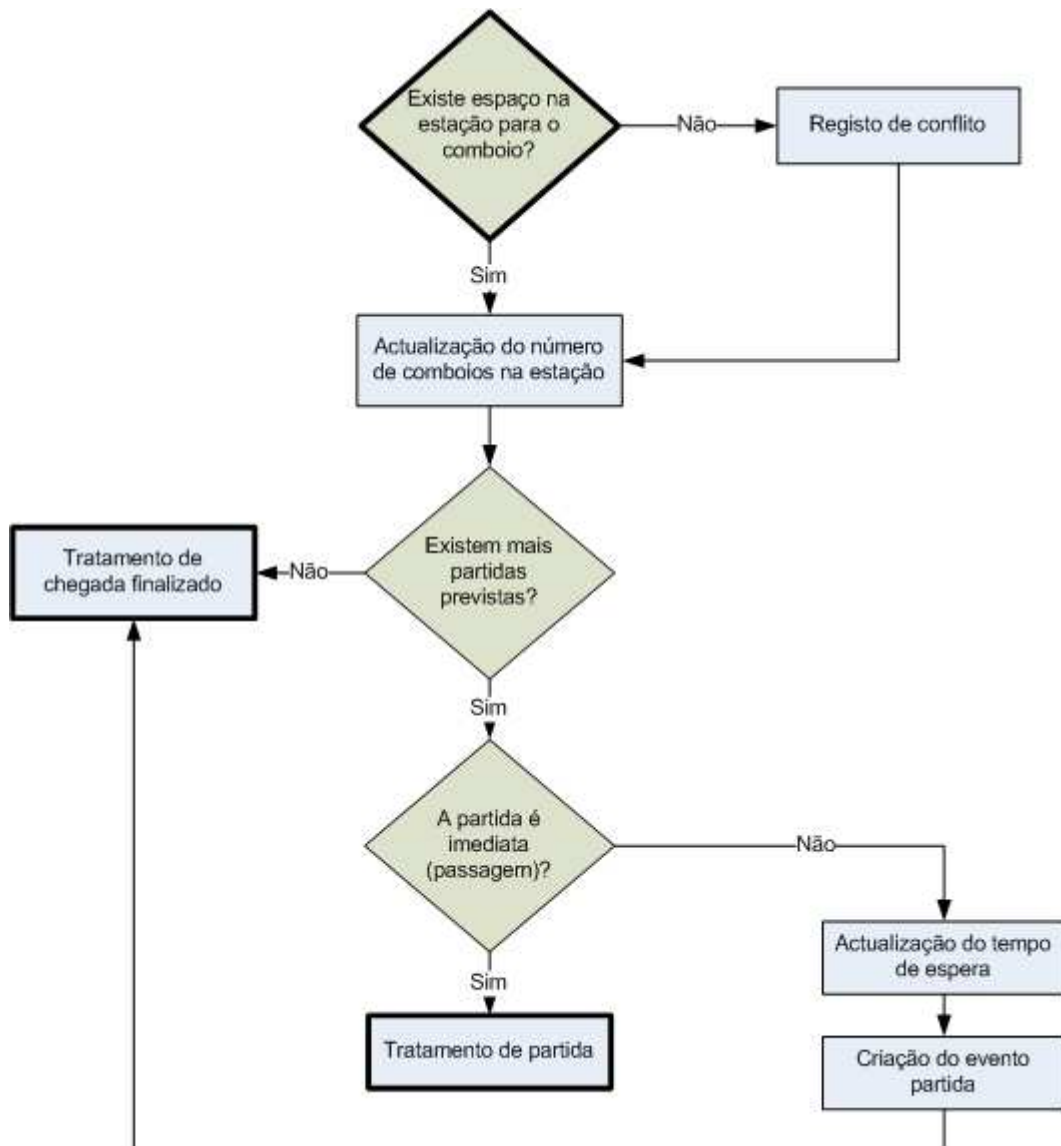


Figura 22 – Processo de tratamento de chegada de uma comboio a uma estação

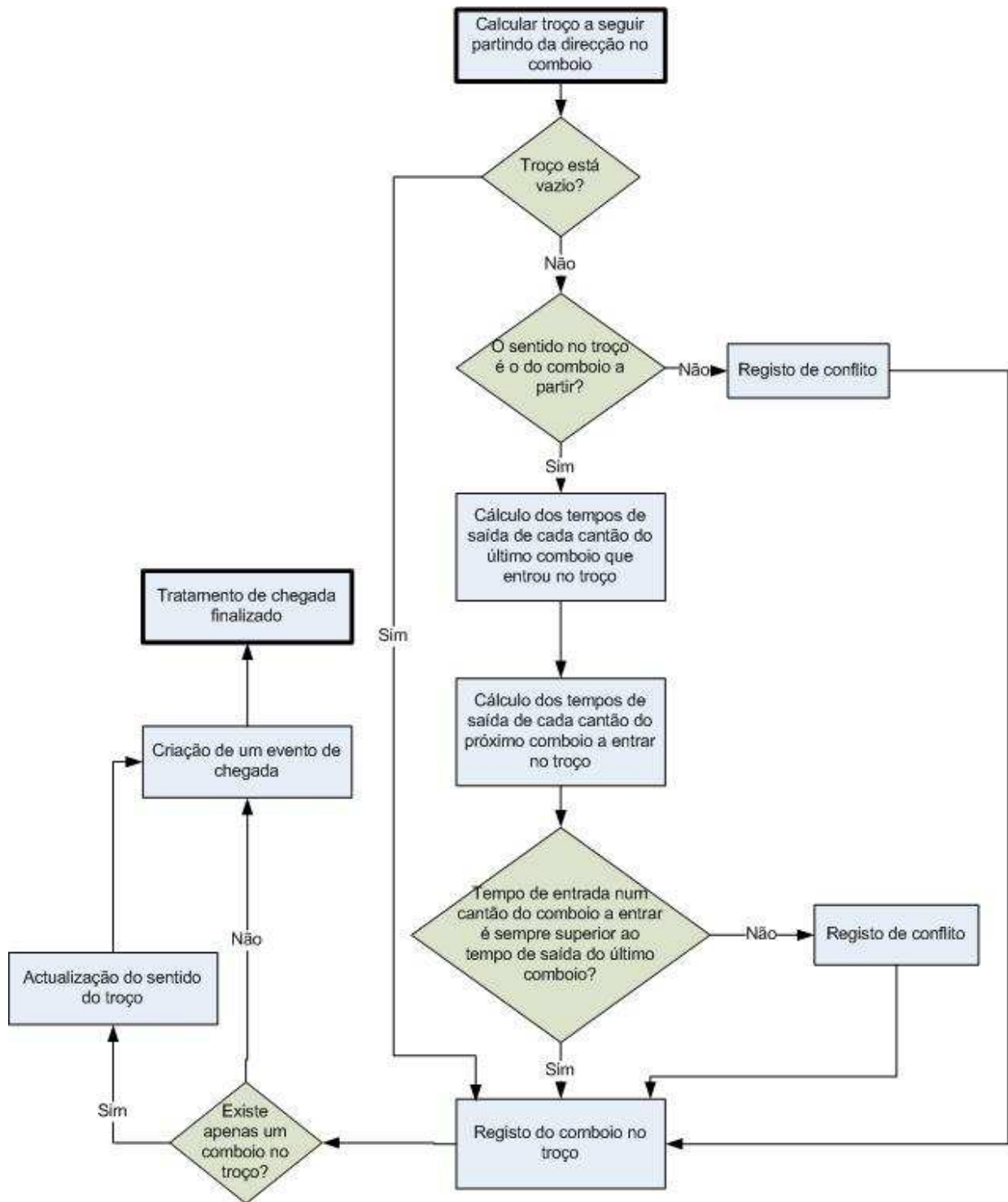


Figura 23 – Processo de tratamento da partida de um comboio

3.3 Descrição Técnica da Aplicação

Nesta secção é feita uma abordagem à tecnologia utilizada (Java) e às suas vantagens e desvantagens face a outras linguagens. É também feita uma análise às opções de projecto tomadas a nível de programação e uma descrição de todas as entidades criadas.

3.3.1 Tecnologia Utilizada

Para o desenvolvimento da aplicação foi escolhido utilizar uma linguagem orientada por objectos, em particular a linguagem de programação Java.

Ao invés de utilizar uma linguagem orientada às funcionalidades, por exemplo C, que obriga a que qualquer actualização passe obrigatoriamente pela alteração da estrutura da aplicação e portanto muito mais complicada e fastidiosa de alterar, as linguagens orientadas por objectos possuem algumas características que tornam estas operações bastante mais intuitivas e facilitadas.

O encapsulamento é talvez a qualidade mais importante. Numa linguagem orientada por objectos existe um agrupamento de ideias relacionadas numa unidade (atributos e funções) que pode ser referenciada por apenas um nome, o que na prática significa que ao definir uma classe (a definição que será utilizada para criar objectos), define-se também atributos que pertencem exclusivamente a essa classe e métodos (na prática funções) também relacionadas apenas com essa classe.

No âmbito do projecto, esta característica traduz-se na criação de entidades independentes e cujo funcionamento interno é transparente para as restantes entidades que com ela se relacionam. Por exemplo, está definido que um comboio que vá partir transmita à estação onde está o sentido que vai seguir e esta diz-lhe para que troço deve seguir. Como é que a estação faz o cálculo é completamente indiferente para o comboio, desde que a saída seja o que está à espera. Aproveitando este exemplo, caso se pretendesse alterar o algoritmo que faz a decisão do troço a seguir, bastava simplesmente alterar o método da classe Estação correspondente à função desejada.

Existem mais algumas características que diferenciam a programação orientada por objectos, tais como:

- Polimorfismo – para uma dada classe podem definir-se vários métodos com o mesmo nome, desde que tenham o tipo de dados de entrada diferentes.
- Herança – esta característica permite a partir de uma classe A, criar uma classe B baseada na classe A e onde se passa a ter disponível todos os atributos e métodos públicos da classe A. Tem-se ainda a possibilidade de adicionar novos atributos e métodos e ainda redefinir os antigos. A herança pode ainda ser simples ou múltipla (herdar de mais de uma classe).

- Possibilidade de representação gráfica – a linguagem de programação Java possui dois packages, o JSWING e o AWT, que permitem a criação de interfaces gráficas. No entanto, para criar uma interface mais complexa, é conveniente utilizar ferramentas de mais alto nível. A nível de apresentação gráfica, mesmo os resultados mais simplistas obtidos são já um avanço para a criação de aplicações bastante mais user friendly.

Estas qualidades das linguagens orientadas por objectos, em particular o JAVA permitem que o código criado seja:

- Uma aproximação do mundo real – a criação de objectos dotados de características próprias e que interagem uns com os outros.
- Extensível - Se no futuro houver necessidades de alterar ou acrescentar algo, as alterações a efectuar no código são bastante mais simples e intuitivas.
- Reutilizável – A herança de classes mais gerais permite adaptar a aplicação a novos requisitos.

A única desvantagem face a linguagens de mais baixo nível é a necessidade de adoptar um conceito de programação um pouco mais complexo e o facto de normalmente o desempenho ser inferior. A diferença de desempenho tem essencialmente a ver com o facto de, depois de um programa em Java ser compilado, ter de ser interpretado pela chamada Java Virtual Machine, desenhada para um sistema em particular, face a uma outra linguagem de programação, por exemplo C, onde a sua compilação nativa tem já uma correlação directa com as instruções do processador. Este problema deriva directamente da portabilidade de JAVA, que graças ao interpretador, permite que o mesmo programa possa facilmente correr em máquinas com processador de tipos diferentes.

3.3.2 Opções de Projecto

Nesta secção é feita a análise a algumas opções de projecto a nível de programação:

Guardar a informação

Quanto a guardar a informação durante o funcionamento da aplicação, como se poderá ver na secção seguinte, a classe principal (a que tem o main) é a classe da interface. Esta contém a lista de Comboios, Eventos, Estações e Log. As Linhas, Troços e Cantões são referenciados nas estações.

Para a criação das listas de Comboios, Estações e Log é utilizada a classe ArrayList que tem a particularidade de funcionar como uma lista dinâmica, sendo possível aceder a um elemento qualquer a partir do valor do índice (inteiro) ou através de um iterador.

A lista de eventos é uma classe criada, sendo designada por EventList e que além de incluir uma ArrayList para eventos como atributo, possui alguns métodos mais complexos para pesquisa e inserção de eventos.

Leitura de Ficheiros

Dos formatos de entrada necessários (Estações, Comboios, Linhas e Horários) apenas foi possível aceder a formato dos Horários real e sendo assim, os formatos dos restantes ficheiros foram criados de modo a tornarem o mais simples possível a leitura dos mesmos. De qualquer modo, como foi referido na secção 3.3.1, as alterações a efectuar para alterar o formato dos ficheiros consistem basicamente em alterar o método de leitura do ficheiro correspondente.

A escolha dos ficheiros é realizada recorrendo a uma componente do package jswing designada por JFileChooser (ver figura 24). Uma particularidade introduzida, é o facto de quando se procura um ficheiro, o browser abrir sempre na directoria onde esteve o último ficheiro carregado.

Dado o formato dos horários fornecido, todas as operações envolvendo tempo são realizadas em segundos.

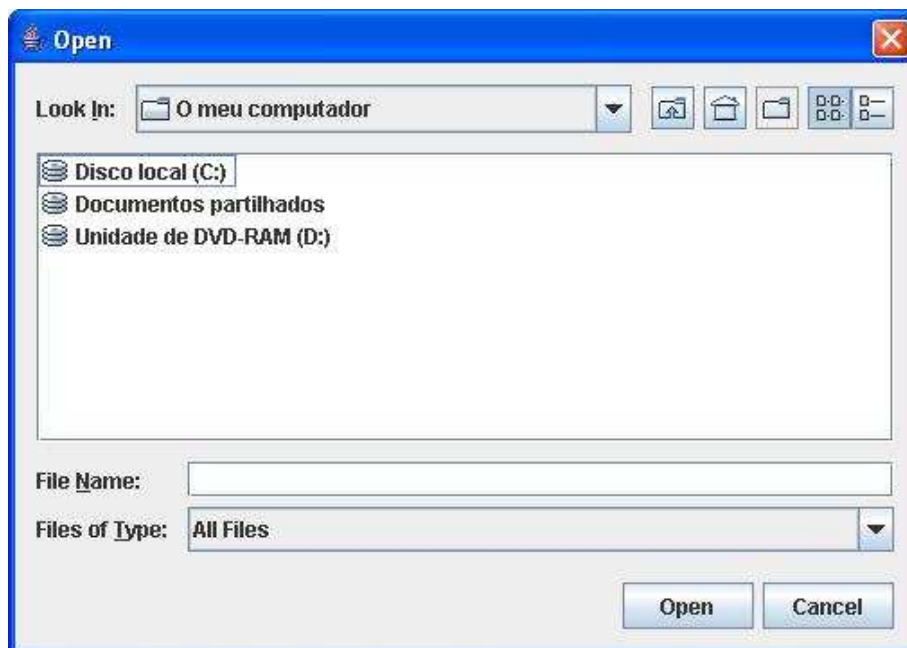


Figura 24 – Ecrã de escolha de ficheiro

Gerador de Gráficos

A geração de gráficos é realizada com recurso a um pacote em JAVA OpenSource de geração de gráficos¹, o resultados que se obtêm não são tão ricos como se gostaria, mas permitem obter uma perspectiva global do horário de um comboio e da relação espaço-temporal dos vários comboios.

A escolha da utilização de um outro software, em vez do desenvolvimento de um gerador de gráficos de raiz, vem do facto de a linguagem de programação JAVA não ter por defeito funções de plot, tornando assim excessivamente moroso a criação do mesmo.

Simulação

A principal opção tomada em relação à simulação foi, como referido na secção 3.1, o facto de se ter passado do cálculo do tempo de viagem utilizando as equações clássicas do movimento, para a utilização do tempo de viagem calculada através da hora de partida e de chegada definidas no horário do comboio. Esta escolha originou também que, para calcular o tempo de viagem num cantão, se use a proporção entre o comprimento do cantão e o comprimentos do troço e o tempo de viagem no mesmo troço.

3.3.3 Principais entidades criadas

Nesta secção descreve-se as diversas classes criadas.

Station

A classe Station representa uma estação. Os métodos que contém são basicamente para fazer a gestão dos comboios que lá se encontram e para indicar o Cantão que cada comboio tem de seguir.

¹ Achim Westermann, aplicação Jchart2D

Train e MathAdvanced

A classe Train representa um comboio, além de guardar os dados do comboio, também guarda os horários associados.

A maioria dos seus métodos são para receber ou devolver atributos. No entanto, contém também os métodos para calcular e actualizar tempos de aceleração, viagem e desaceleração, para serem usados quando se pretender utilizar as equações do movimento para calcular tempo de viagem.

A classe Train contém ainda os métodos utilizados para a verificação de conflitos do tipo dois comboios no mesmo cantão.

A classe MathAdvanced está relacionada com os cálculos desses mesmos tempos e outras variáveis de interesse.

Troco e Segment

A classe Troco representa um troço e, além de guardar as suas características, possui métodos para a gestão de comboios que entram e saem. A classe Troco contém um inteiro indicador do sentido adoptado, um contador do número de comboios e a referência ao último comboio que entrou. Estes dados permitem detectar os conflitos do tipo: dois comboios no mesmo troço com sentidos opostos. É ainda na classe Troco que é feita a verificação de existência de conflitos do tipo Pass.

A classe Segment corresponde aos Cantões, guarda nos atributos as suas características e os métodos que contém permitem devolver atributos e registar a entrada e saída de comboios.

Dados e Log

A classe Dados é responsável pela leitura e carregamento dos ficheiros de input e pela gravação dos horários alterados e log.

Os seus métodos tipicamente recebem uma referência para um ficheiro e de seguida, no caso das leituras, lêem e interpretam para o formato correspondente. No caso das escritas, gravam nos ficheiros os dados pedidos.

TimeBlock, TimeBlockPlus e HorarioTrain

As classes TimeBlock e TimeBlockPlus são uma representação interna dos horários de cada comboio, incluem já o tempo de viagem e a direcção. A classe TimeBlockPlus inclui ainda uma referência para o objecto que representa a estação correspondente.

A classe HorarioTrain basicamente guarda a informação segundo o formato do ficheiro de horários original. Esta representação é depois utilizada na criação dos gráficos dos horários e na detecção de alterações introduzidas pelo utilizador.

Todos os objectos destes tipos são criados na leitura do ficheiro de horários.

Event e EventList

A classe Event representa um evento, enquanto que EventList é uma lista de eventos que inclui métodos para pesquisa e inserção de eventos.

Na leitura dos dados dos horários são criados os primeiros eventos. Durante a simulação são criados eventos sempre que existirem partidas e chegadas.

Simulator

A classe simulador representa o simulador propriamente dito. Permite correr a aplicação possuindo também métodos específicos para tratar eventos de partida e de chegada.

Interface

A classe interface é a classe que tem o método Main e portanto será a classe principal. Contém toda a estrutura para criar e configurar o design gráfico e métodos para processar cada um dos botões e opções disponíveis.

É na classe Interface que é criada a classe Dados para o carregamento dos ficheiros e a classe simulador para correr as simulações.

A classe Interface é assim responsável por toda a interacção com o utilizador.

4. Conclusões Finais

A lista de todos os problemas possíveis de acontecer no dia-a-dia de uma rede de comboios é longa e variada. Existem imensos aspectos passíveis de sofrer falhas, humanas ou não. Tendo em conta que, em particular nos comboios urbanos, podem estar envolvidas muitas vidas humanas, os cuidados na prevenção e resolução de problemas têm de ser redobrados.

Muitos dos desafios existentes na gestão de tráfego ferroviário provêm de actividades que dependem da coordenação de imensas entidades, desde a gestão da carga e descarga de mercadorias e passageiros até à alocação dos vários serviços, de modo a poder oferecer o melhor serviço possível.

A gestão de tráfego ferroviário em linhas de baixo tráfego é apenas um dos muitos problemas existentes, mas que existe e tem de ser confrontado diariamente.

A presente tese constitui, em particular em Portugal, um passo significativo para uma melhoria dos actuais serviços existentes. A aplicação criada não resolve todos os problemas, mas fornece um forte ponto de partida para a criação de uma ferramenta mais profissional e completa.

A ferramenta desenvolvida acaba por ter realmente potencial para se tornar numa ferramenta de apoio à decisão, permitindo nesta fase:

- Carregar a descrição de estações, linhas, comboios e horários;
- Introduzir alterações nos horários dos comboios, alterações essas que podem ser representativas de conflitos ou de soluções;
- Simulação dos horários introduzidos/alterados, prevendo os possíveis conflitos;
- Exportação dos horários alterados e do resultado da simulação.

O que acaba por ser um resultado bastante positivo para as expectativas iniciais.

4.1 Melhorias possíveis de introduzir numa futura versão da aplicação

Existem vários aspectos possíveis de alterar e introduzir nas diversas áreas da aplicação desenvolvida.

Gestão dos horários dos comboios:

- Introdução de opções para alterações mais complexas aos horários, tanto para a introdução de conflitos, como para a introdução de soluções, em particular, templates de soluções “típicas”, passíveis de serem customizados, e alteração simultânea a vários horários e a vários comboios;
- Possibilidade de introduzir soluções baseadas em rerouting de comboios, ou seja, que permitam soluções envolvendo mudanças para linhas ou destinos alternativos.

Apresentação de resultados de simulação:

- Introdução de uma representação gráfica mais rica e interactiva, em particular uma apresentação mais detalhada no que se refere à informação das linhas, estações e comboios.
- Introdução de soluções directamente na representação gráfica, manipulando as linhas representativas das deslocações dos vários comboios. Esta opção seria no entanto muito complexa.
- A atribuição de um custo a cada solução introduzida. Este custo estaria dependente essencialmente do tipo de comboio, da altura do dia em que as alterações seriam introduzidas e da quantidade de alterações a efectuar.

Gestão da informação na aplicação:

- Introdução de um histórico de conflitos e soluções introduzidas. Este histórico permitiria não só comparar diversas soluções, mas também fazer um “rollback” às diversas alterações.
- Introdução de conectividade com outras aplicações de gestão ferroviária para uma utilização mais rápida e eficaz dos resultados obtidos.

4.2 Dificuldades durante o desenvolvimento

A principal dificuldade durante o desenvolvimento deste projecto foi a criação de um modelo para a simulação. Modelo esse que teria que cumprir rigorosamente uma série de critérios de modo a que o resultado final fosse, dentro das limitações existentes, o mais fidedigno possível para com a situação real.

Outras dificuldades relevantes foram a definição das opções da interface e a sua interacção com o utilizador, ou mesmo a criação dessa mesma interface dadas as ferramentas disponibilizadas e o conhecimento técnico existente. De notar o enriquecimento pessoal adquirido na utilização de linguagens de programação orientadas por objectos, cujo conhecimento no início do desenvolvimento da tese era praticamente inexistente.

No decorrer do desenvolvimento da tese foi colocada a possibilidade de criar um modelo que propusesse uma solução para a resolução de conflitos. Na prática, seria um modelo representativo das linhas, comboios e horários acompanhado de um conjunto de restrições representativas dos conflitos a resolver e uma função de custo, dependente das alterações que constituem a solução, a otimizar.

O desenvolvimento de tal modelo e da sua optimização constitui, por si só, material suficiente para a elaboração de uma outra tese, de tal modo que não houve disponibilidade temporal para chegar a uma solução válida.

Contudo, foi possível detectar um grande problema na criação do modelo: a dificuldade na criação de restrições que abrangessem todos os dados, ou seja, as restrições são sempre relativas.

Um exemplo claro do problema é pensar que para um dado instante, um comboio X apenas entra num determinado cantão quando este se encontra vazio, o que pode acontecer em apenas duas situações:

- A hora de entrada de um comboio que passe nesse cantão ainda não chegou;
- A hora de saída de um comboio que passe nesse cantão já passou.

A primeira restrição apenas é válida para os comboios que ainda não entraram no cantão e a segunda para aqueles que já saíram, fazendo assim com que cada uma das restrições apenas se aplique a um grupo de comboios. Com a agravante de que esse grupo é diferente para cada comboio e ainda que esse mesmo grupo pode variar com o tempo para o mesmo comboio, basta pensar que um comboio pode ser ultrapassado numa estação.

4.3 Trabalhos futuros

Além das melhorias sugeridas na secção 4.1, o facto da criação do simulador fazer parte de uma ideia mais abrangente para a resolução de conflitos de tráfego ferroviário em linhas de baixo tráfego, sugere logo à partida um seguimento do trabalho desenvolvido.

O trabalho seguinte seria a construção de um modelo, do qual foi feita referência na secção anterior, para a resolução de conflitos e o seu algoritmo de optimização, de modo a que se construa uma ferramenta que proponha uma solução para um conjunto de conflitos.

Tendo uma solução completa para a resolução de conflitos, o passo seguinte acaba por ser a reestruturação do processo, de modo a permitir aplicar a ferramenta em linhas de médio e alto tráfego, em que a dificuldade resulta directamente da complexidade e densidade das linhas existentes.

Num futuro a longo prazo, o objectivo é a criação de um sistema que interaja directamente com os comboios, recebendo actualizações da posição dos mesmos, detectando conflitos, produzindo soluções e alterando directamente os horários dos comboios envolvidos com a melhor solução possível.

Referências Bibliográficas

- [1] Ahuja, R. K., A Decision Support System for Service Design, Apresentação da Innovative Scheduling, www.innovativescheduling.com, 2006.
- [2] Pranzo, M, D'Ariano, Pacciarelli, D, The effectiveness of static implications in real-time railway traffic management, Advanced OR and AI Methods in Transportation, 2007.
- [3] D'Ariano, A., Modeling reordering and local rerouting strategies to solve train conflicts during rail operations, Transport & Planning, TU Delft, 2006.
- [4] Higgins, A., Ferreira, L., Kozan, E., Modelling single line train operations, Transportation Research Record 1489, Journal of the Transportation Research Board, 1995.
- [5] Missikoff, M., An Object Oriented Approach to an information and decision support system for railway traffic control, Knowledge-Based Intelligent Electronic Systems, 1997. KES apos;97. Proceedings., 1997 First International Conference on, 1997
- [6] ICT Results,
<http://cordis.europa.eu/ictresults/index.cfm/section/news/Tpl/article/BrowsingType/Features/ID/78084>, 2005.
- [7] Eckel, B., Thinking in JAVA, Prentice Hall, 4ª edição, 2006.
- [8] Paul S. Wang, JAVA with Object-Oriented Programming, Thomson Brooks/Cole, 2º edition, 2003
- [9] R. G. Crespo, Acetatos teóricos de Programação por Objectos LEEC/MEEC 2005/2006

Anexo A - Manual de Utilização

Esta secção contém toda a informação necessária para a utilização da aplicação desenvolvida, desde o formato dos ficheiros de input até às diversas opções disponíveis, passando pela descrição dos passos necessários para realizar simulações e testar soluções.

A1. Ficheiros de Input

Existem quatro formatos de ficheiros para input, sendo que apenas os dados dos horários têm o formato utilizada actualmente pelas entidades gestoras do tráfego ferroviário. A extensão oficial dos ficheiros é .DAT

Os ficheiros de dados de estações, comboios e linhas têm um parâmetro por linha enquanto que o ficheiro dos horários tem em cada linha a informação completa de identificação do comboio e da informação das horas de partida e de chegada a cada estação.

Tabela 1 – Formato dos dados do ficheiro de estações

parâmetros estação 1	Id da estação	(string)
	nome	(string)
	tipo de estação	(string)
	número máximo de comboios	(int)
	tamanho (metros)	(int)
...
parâmetros estação n	Id da estação	(string)
	nome	(string)
	tipo de estação	(string)
	número máximo de comboios	(int)
	tamanho (metros)	(int)

Tabela 2 – Formato dos dados do ficheiro de comboios

parâmetros comboio 1	Id do comboio	(string)
	nome	(string)
	tipo de comboio	(string)
	velocidade máxima	(int)

	aceleração máxima	(int)
	desaceleração máxima	(int)
...
parâmetros comboio 2	Id do comboio	(string)
	nome	(string)
	tipo de comboio	(string)
	velocidade máxima	(int)
	aceleração máxima	(int)
	desaceleração máxima	(int)

Tabela 3 – Formato dos dados do ficheiro de linhas

parâmetros troço 1	Id da primeira estação	(string)
	Id da última estação	(string)
	comprimento do troço (metros)	(int)
	número de cantões	(int)
cantão 1	comprimento do cantão (metros)	(int)
	velocidade média (km/h)	(float)
...
cantão n	comprimento do cantão (metros)	(int)
	velocidade média (km/h)	(float)
...
parâmetros troço n	Id da primeira estação	(string)
	Id da última estação	(string)
	comprimento do troço (metros)	(int)
	número de cantões	(int)
cantão 1	comprimento do cantão (metros)	(int)
	velocidade média (km/h)	(float)
...
cantão n	comprimento do cantão (metros)	(int)
	velocidade média (km/h)	(float)

Tabela 4 – Formato de dados do ficheiro de horários

	Indicador tipo de dados	ID comboio	Código regime	Data Validade		
cabeçalho horário	"1"	dígitos 2-7	8-11	AAAAMMDD		
	Indicador tipo de dados	ID estação	Hora chegada	Hora partida	Indicador de direcção	Linha do troço
ponto operacional	"2"	2-8	HHMMSS	HHMMSS	"1"-ascendente, "2"-descendente	dígito 22

A2. Acções disponíveis

Descrição das diversas operações possíveis na aplicação.

Carregar dados de entrada

Seleccionar "Ficheiros" e escolher os dados a carregar. (ver figura 25)

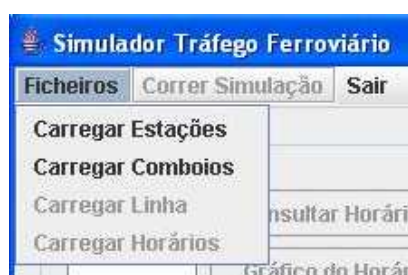


Figura 25 – Menu de carregamento de ficheiros

É aberta uma janela de browser. Seleccionar o ficheiro pretendido e confirmar (ver figura 26)

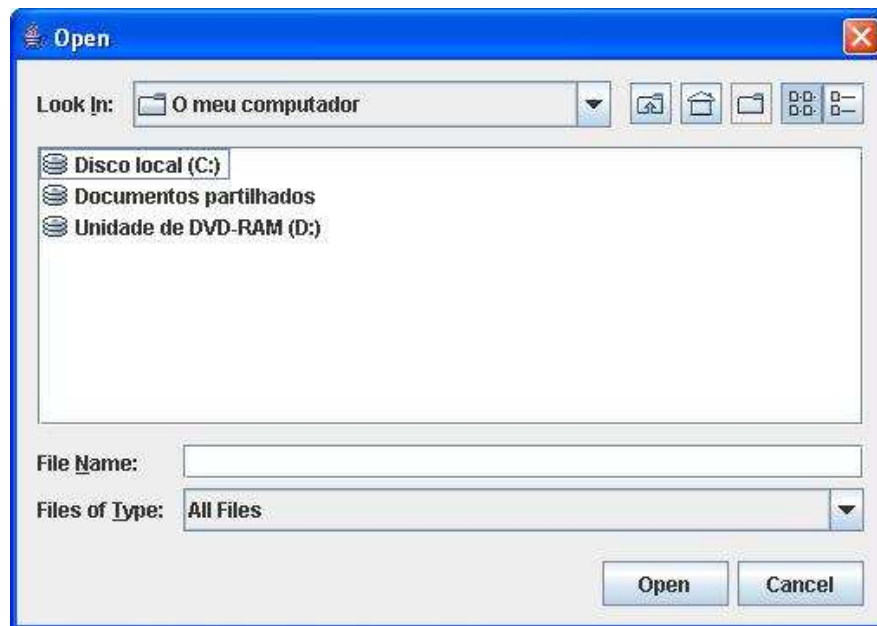


Figura 26 – Janela para procura de ficheiro

Os ficheiros têm de ser carregados por uma ordem pré-definida: Estações->Comboios->Linha ->Horários, sendo a ordem de carregamento forçada através da aplicação.

Por exemplo, enquanto os comboios não forem carregados, a opção de carregar horários está bloqueada. Quando já se fez todos os carregamentos e se volta a fazer por exemplo o carregamento dos comboios, a aplicação obriga a introduzir novamente as linhas e os horários.

Visualização do horário de um comboio

Depois de ter carregado todos os ficheiros, na secção “Comboio” as opções de visualização passam a estar disponíveis. “Consultar horário” preenche, a tabela na secção “Horários”, com o horário do comboio seleccionado. “Gráfico do Horário” gera uma representação gráfica do mesmo (ver figura 27).

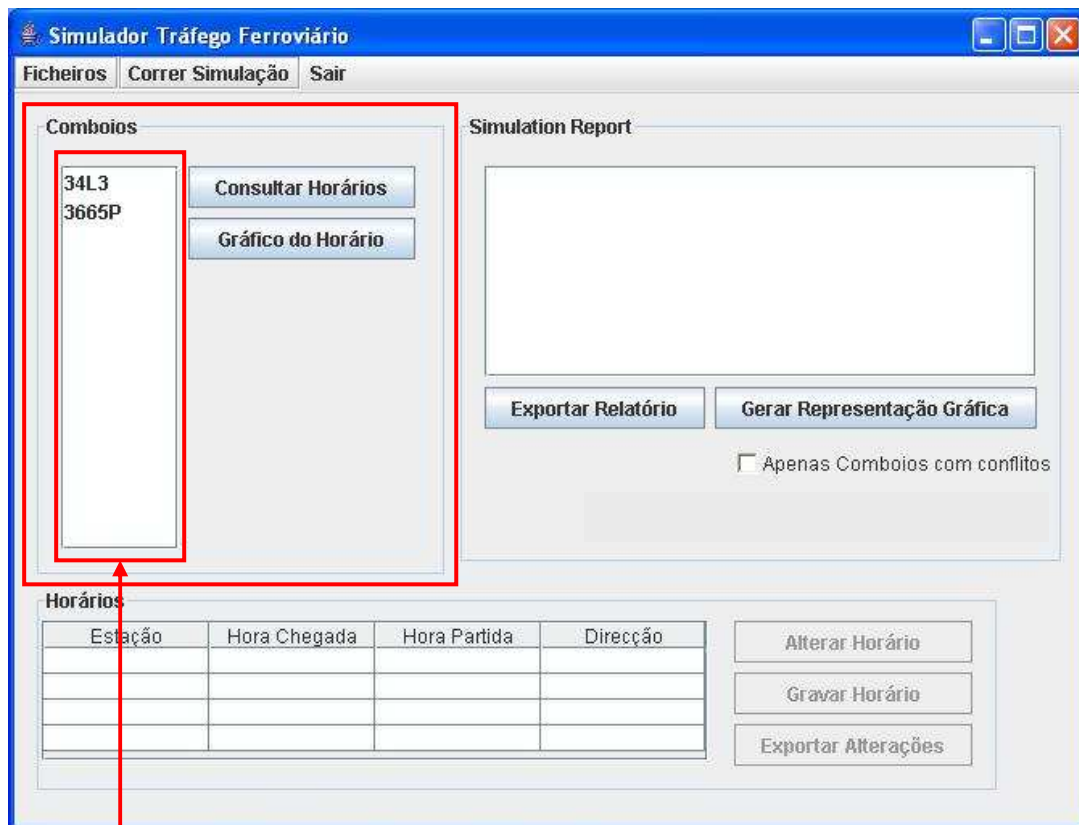


Figura 27 – Ecrã principal, selecção da área de comboios

Lista dos vários
comboios carregados

Introdução de conflitos/soluções

A introdução de conflitos e de soluções na prática consiste em fazer alterações ao comboio pretendido. Tendo os ficheiros carregados, seleccionar um comboio e pressionar “Consultar Horários”. Alterar as “Horas de Chegada” e “Horas de partida” pretendidas e seleccionar “Alterar Horário”.

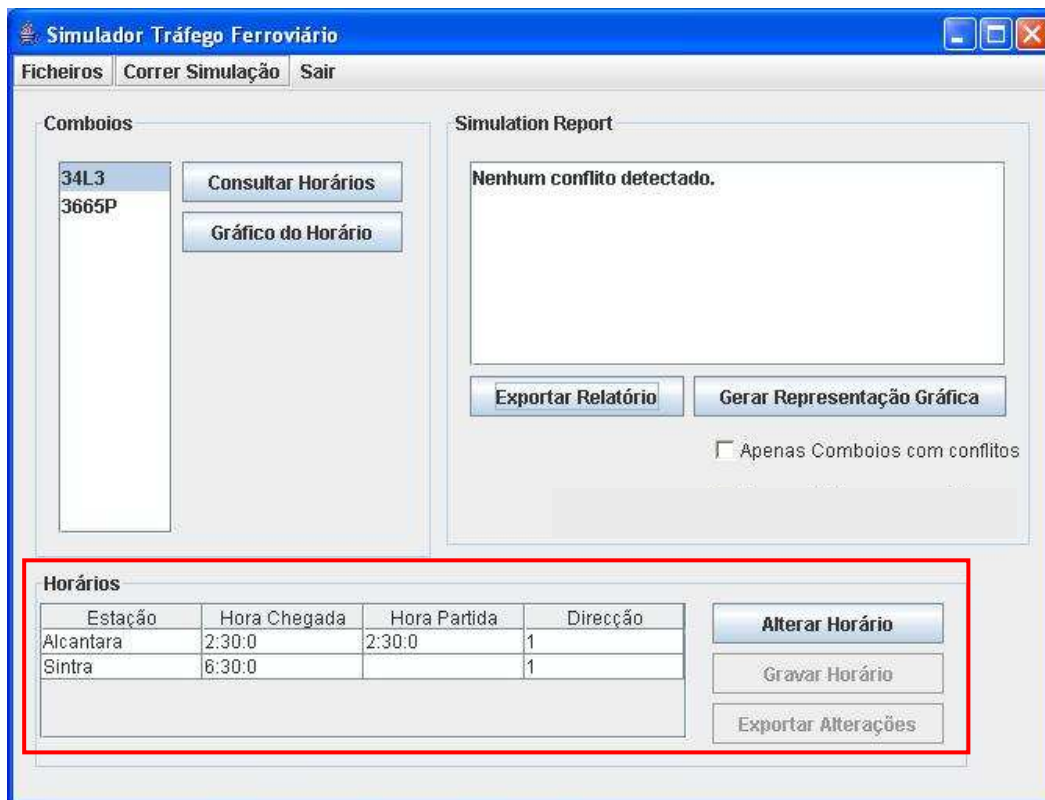


Figura 28 - Ecrã principal, selecção da área de horários

Gravar e exportar alterações nos horários para um ficheiro

Depois de ter realizado alguma alteração a horários, "Gravar Horário " permite gravar, num ficheiro, o horário com as alterações introduzidas. O ficheiro gerado tem a formatação do ficheiro de horários original.

"Exportar Alterações" abre uma janela de browser para escolher o nome do ficheiro e a directoria onde guardar um registo das alterações efectuadas. A formatação do ficheiro criado pode ser visualizada na tabela 4.



Figura 29 - Ecrã principal, selecção das opções de gravação e exportação de alterações

Efectuar Simulação

Depois de todos os ficheiros estarem carregados, a opção “Correr Simulação” passa a estar disponível. Depois da simulação feita, o ecrã “Simulation Report” apresenta os conflitos previstos.

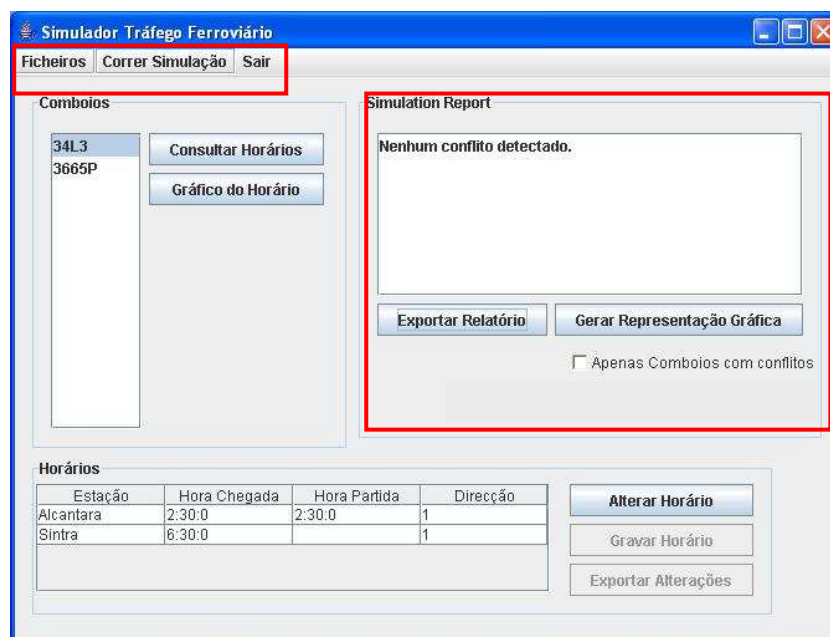


Figura 30 - Ecrã principal, selecção da opção de iniciar simulação e simulation report

Tabela 5 – Mensagens de relatório de conflitos

Formato das mensagens de report
"Nenhum conflito detectado"
hh:mm:ss "Comboio" idComboio "sai de" nomeEstação "para um troço com sentido contrário em direcção a" nomeEstação
hh:mm:ss "Comboio" idComboio "entra num cantão ocupado entre" nomeEstação "e" nomeEstação
hh:mm:ss "Comboio" idComboio "entra na estação sobrelotada de" nomeEstação

Visualização dos horários dos comboios após simulação

Depois da simulação estar concluída a opção “Gerar Representação Gráfica” passa a estar disponível. Através da checkBox pode gerar a representação apenas com os comboios com conflitos.

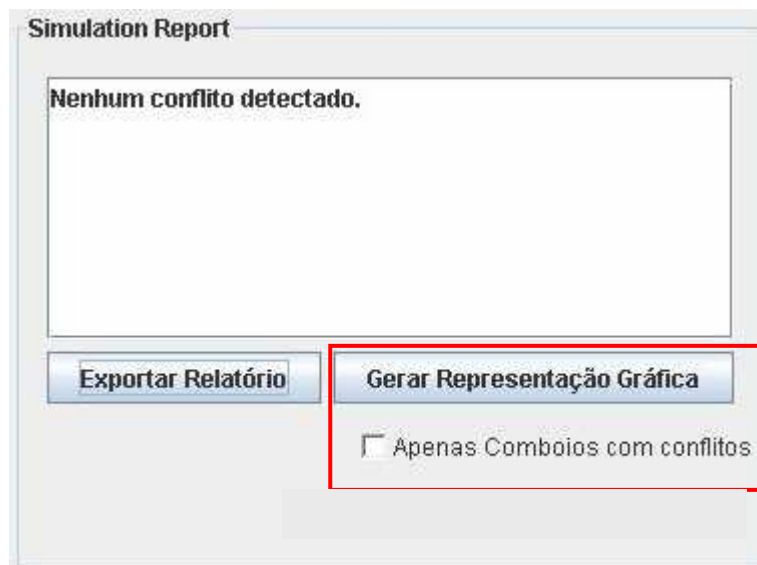


Figura 31 – Ecrã de simulation report, selecção de geração de gráficos

Exportar resultados da simulação

Depois de efectuada a simulação é possível Exportar o resultado obtido. Seleccionar “Exportar Relatório”, é aberta uma janela para seleccionar o nome do ficheiro e a directoria onde gravar.

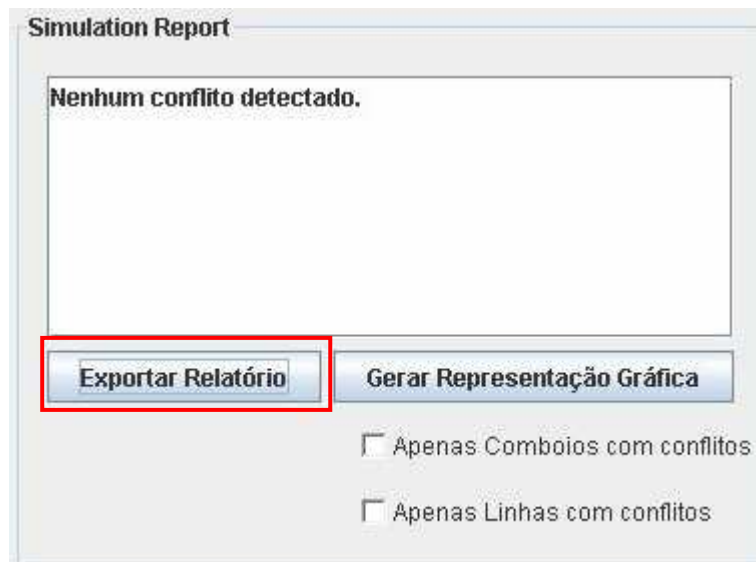


Figura 32 – Ecrã de simulation report, selecção de opção para exportação

Anexo B – UML da aplicação

Os diagramas UML – *Unified Modeling Language* – permitem uma melhor compreensão do modelo e estrutura da aplicação criada. Na figura 33 encontram-se definidas as relações, entre as várias classes criadas. Nas figuras seguintes estão as representadas cada classe em detalhe.

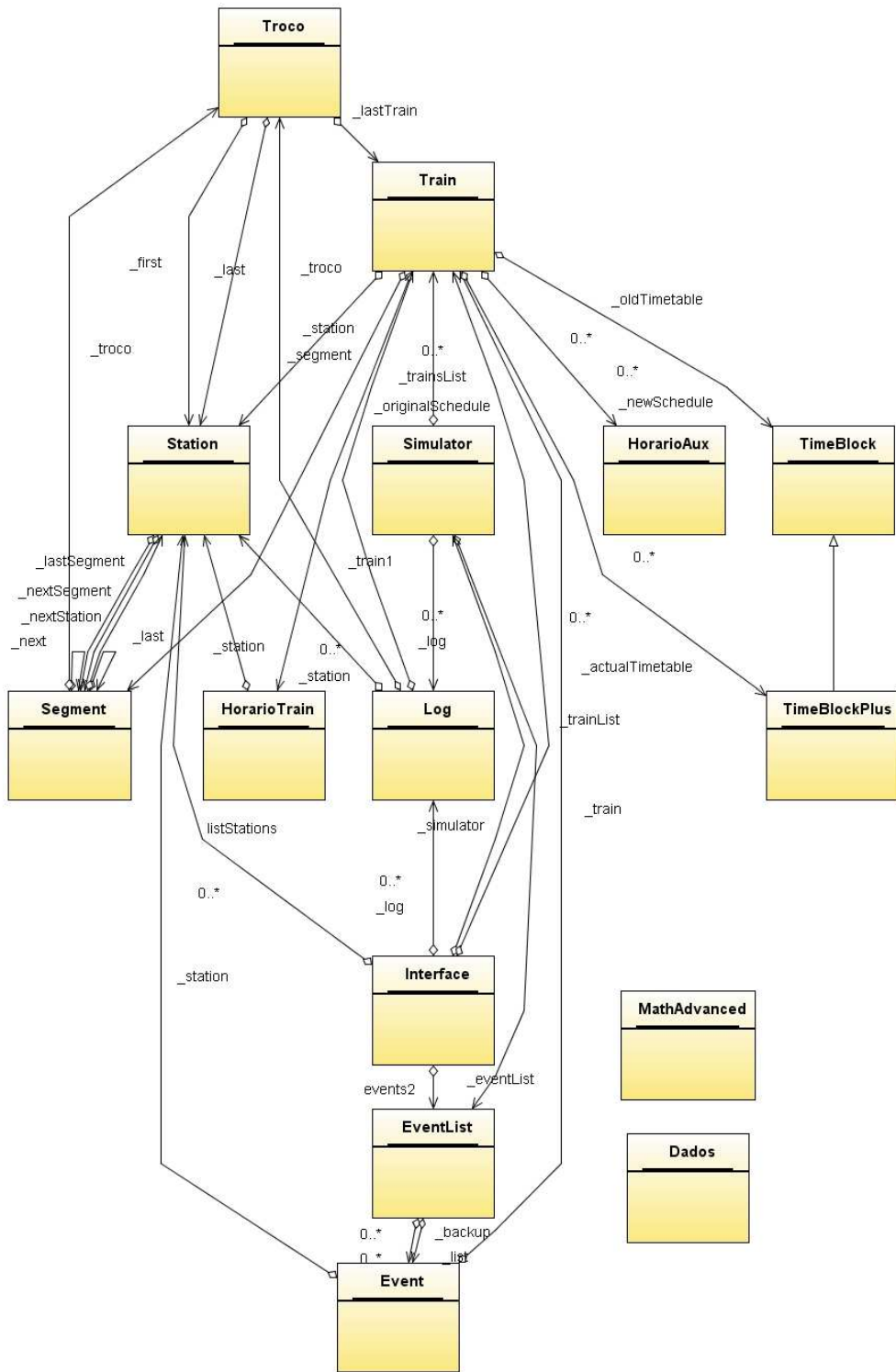


Figura 33 - Diagrama UML da aplicação desenvolvida

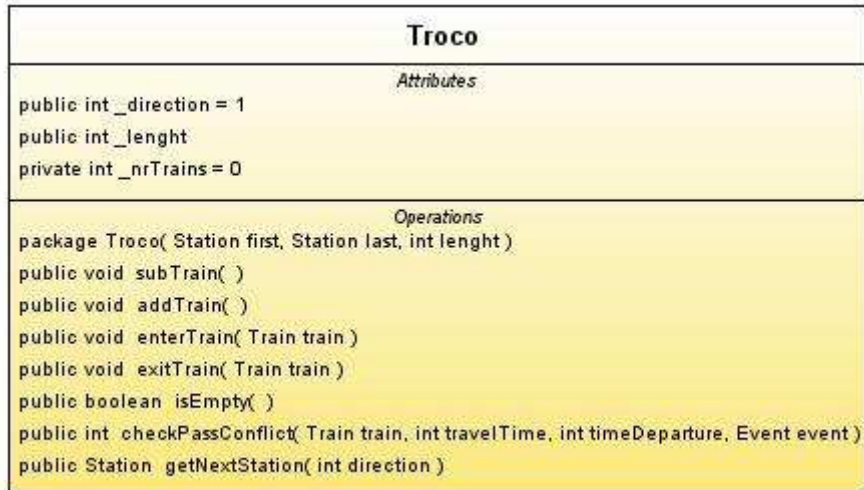


Figura 34 – UML da classe Troco



Figura 35 – UML da classe Station

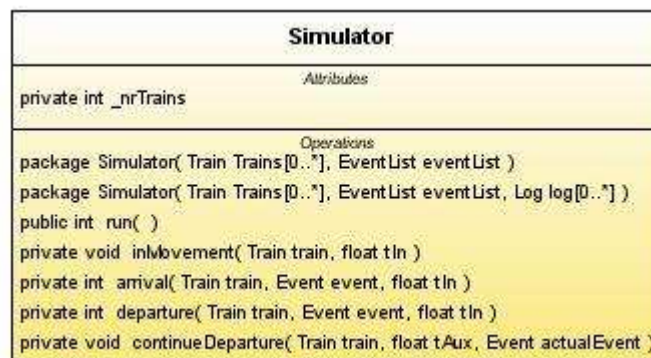


Figura 36 – UML da classe Simulator



Figura 37 – UML da classe Train

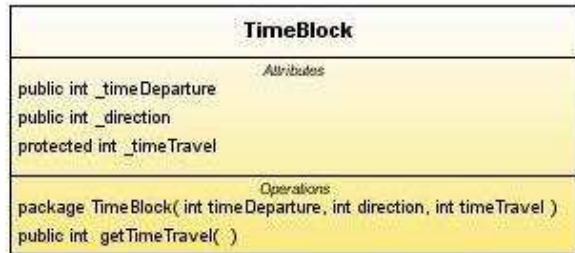


Figura 38 – UML das classes HorarioAux e TimeBlock



Figura 39 – UML da classe TimeBlockPlus

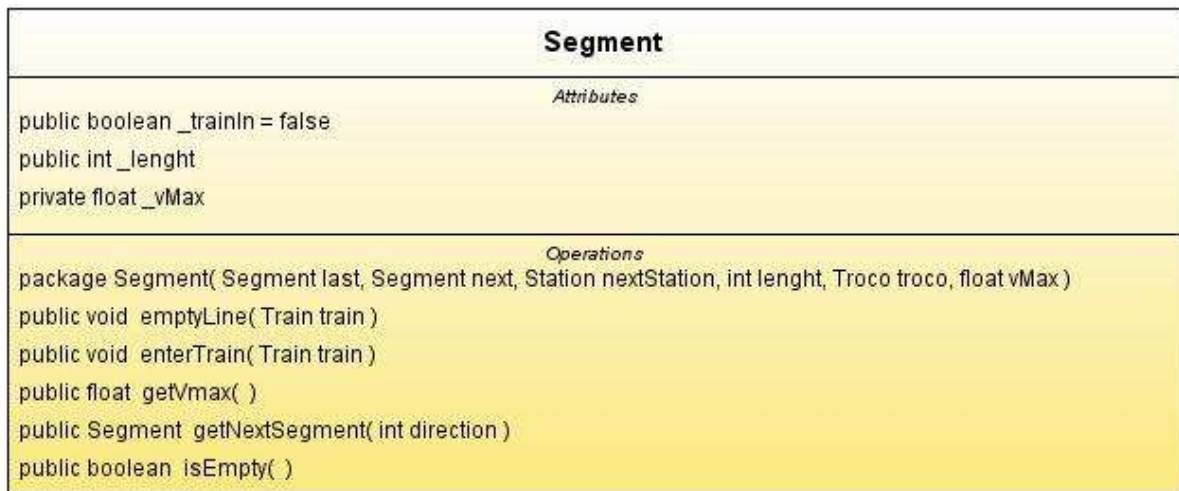


Figura 40 – UML da classe Segment

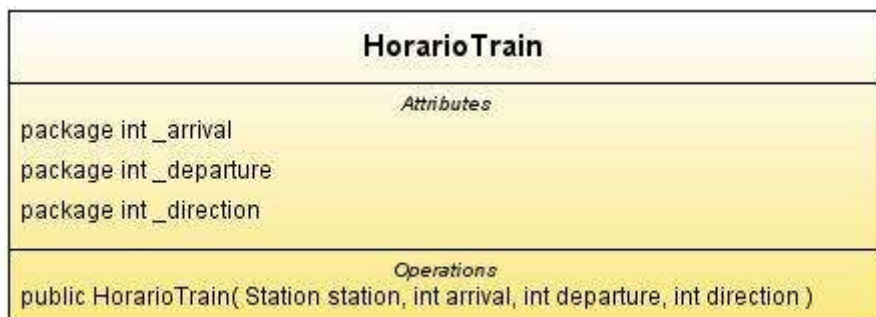


Figura 41 – UML da classe HorarioTrain

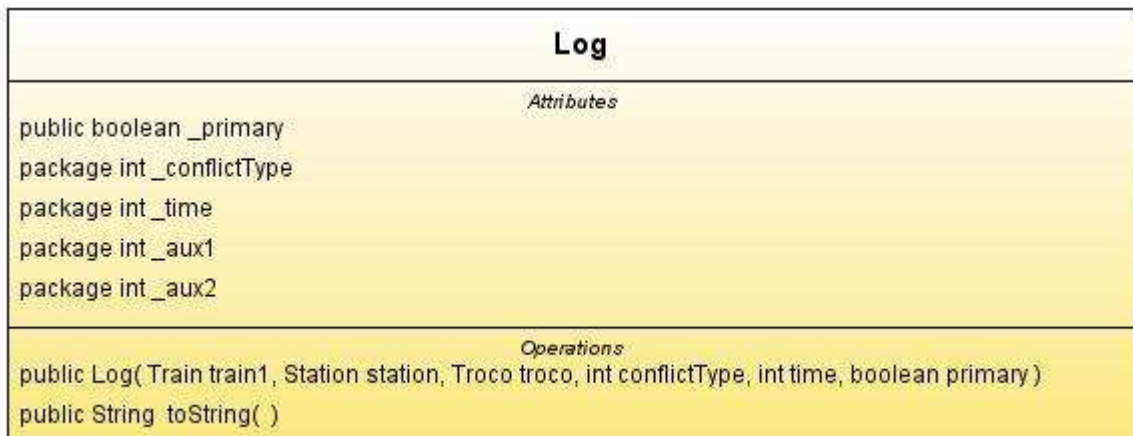


Figura 42 – UML da classe Log

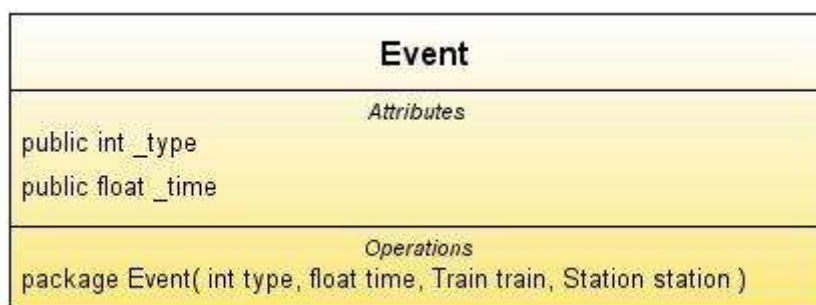


Figura 43 – UML da classe Event



Figura 44 – UML da classe EventList

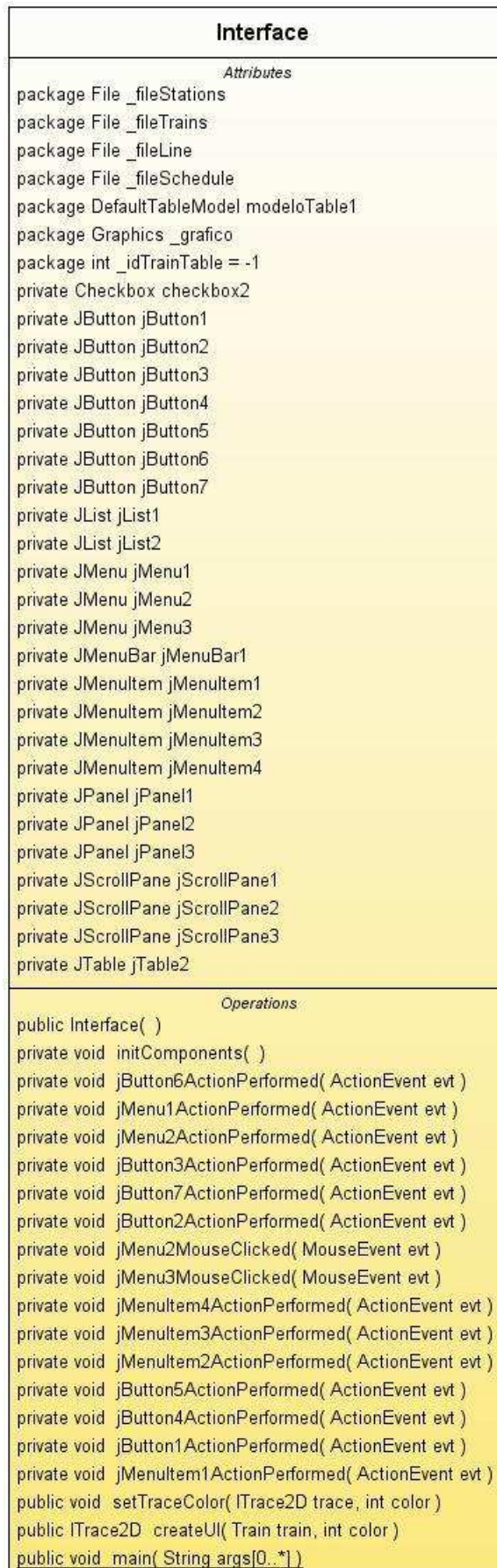


Figura 45 – UML da classe Interface



Figura 46 – UML da classe Dados

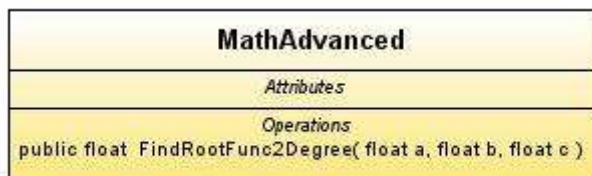


Figura 47 – UML da classe MathAdvanced