

Exercício 9 - com MS-Access

Este exercício é uma extensão do anterior e tem por objetivos:

- Construir uma rede de arcos e nós; e, a partir dela,
- Construir os inputs (entradas) requeridos para um programa de cálculo de caminhos mínimos desenvolvido em Object Pascal, originalmente na plataforma Delphi 6, da Borland Inc., depois para Delphi Rad Studio 10, da Embarcadero.

9.1 Construção da rede metro-ferroviária a partir da tabela **MLS_od07**, como um grafo de arcos e nós

Na **Aula 7 com MS-Access**, o arquivo **MetroLineStations** foi importado para um BD e depois filtrado para **MLS_od07** para refletir a da situação existente em **31.dez.2006** e, a partir de uma lista de pontos, indicativos de estações de cada linha. Depois foram geradas as sequências de trechos interligando estações vizinhas, sucessivamente, segundo uma das direções, em cada linha.

Como exercício extra, pediu-se que fossem geradas as sequências inversas de cada linha, ou seja dos percursos na direção oposta. Cada trecho, unindo duas estações vizinhas de uma linha, em uma das direções, constitui um arco da rede, na tabela **ARCOS_od07**.

Para completar tal rede, há de se criar também os arcos de transbordo, que permitam modelar as transferências entre linhas feitas em algumas estações, segundo as necessidades de viagem dos passageiros.

Criar arcos de transbordo nas estações operadas por mais de uma linha

- (i) Os arcos de transbordo acontecem nos casos de mesma estação (CODE, NOME, NAME iguais) e linhas ou suas extensões diferentes (LINE diferente);
- (ii) Os arcos de transbordo devem considerar como possibilidades ambas as direções da linha para a qual se iria embarcar – para isso foram adicionadas as colunas DIRECAOo e DIRECAOd; a Figura 1 esquematiza as linhas, suas direções e arcos de transbordo, para os casos, por exemplo, das estações Ana Rosa e Paraíso;
- (iii) A composição (LINEo, CODEo, CODEd) não mais pode ser usada como chave candidata. Nos arcos de transbordo CODEo = CODEd. E ainda que LINEo ≠ LINEd, haveria dois arcos, (... , "O", "I") e (... , "I", "O"), para cada composição (LINEo, CODEo, LINEd, CODEd). Ela necessita ser estendida para (LINEo, CODEo, CODEd, LINEd, DIRECAOo, DIRECAOd).
- (iv) O número de arcos em cada estação crescerá geometricamente ($L \times 2^L$, para $L > 1$, onde L é o número de linhas) com o número de linhas operando nessa estação. Para duas linhas, seriam criados 8 (oito) arcos de transbordo; para três linhas, seriam

criados 24 arcos de transbordo; para quatro linhas, seriam criados 64 arcos de transbordo;

- (v) Uma alternativa é de reduzir os arcos pela introdução de uma simplificação, ou seja, da criação de um nó fictício em cada estação, de forma que apenas haja apenas um par de arcos por direção de cada linha que passa na estação para esse nó. No que o número de arcos aumenta linearmente com o número de linhas, ou seja, $(4 \times L)$. Para duas linhas, resulta 8 (oito) arcos, o mesmo do caso não simplificado; mas para três linhas, seriam 12 arcos e não 24; e para quatro linhas, seriam 16 arcos e não 64;

Caso em que não haveria arcos com $(\dots, DIRECAOo, DIRECAOd) = (\dots, "O", "O")$, $(\dots, "O", "I")$, $(\dots, "I", "I")$ ou $(\dots, "I", "O")$, mas apenas $(\dots, DIRECAOo, DIRECAOd) = (\dots, "O", "X")$, $(\dots, "X", "O")$, $(\dots, "I", "X")$ ou $(\dots, "X", "I")$ – conforme esquematizado na Figura 2;

Figura 1 – Arcos de percurso de duas linhas e de transbordo entre elas – exemplo para o caso de duas linhas

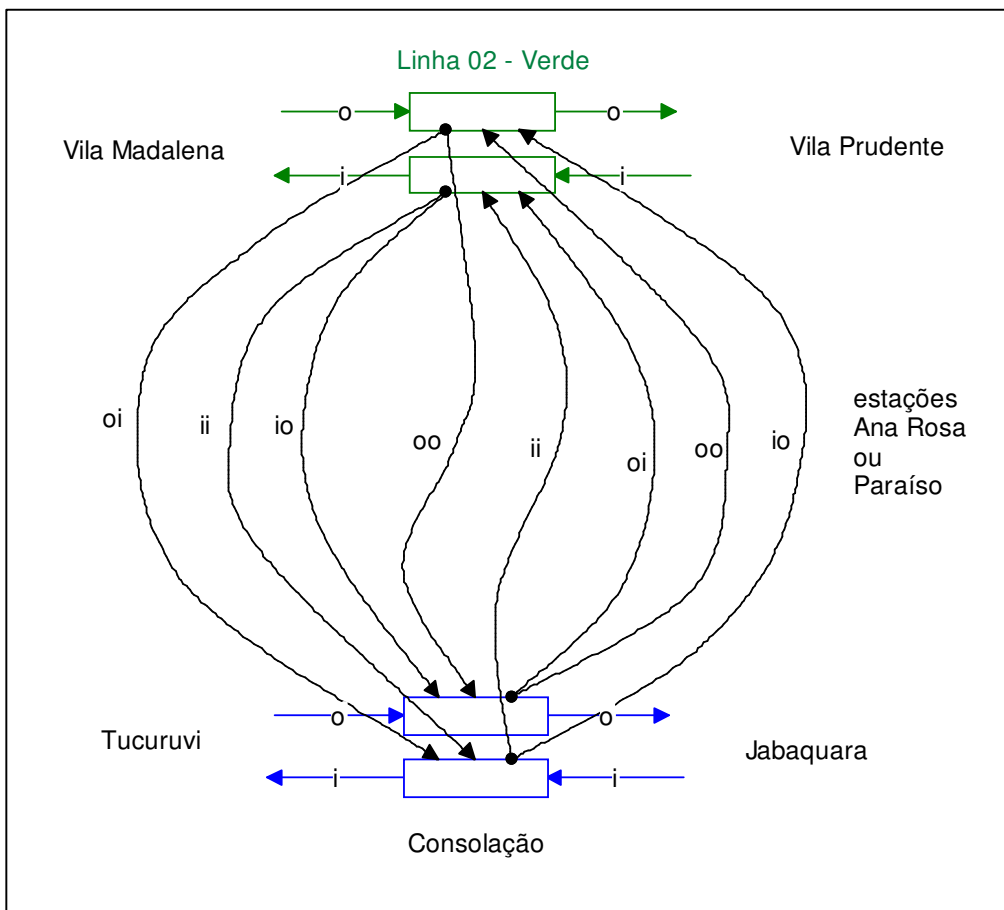
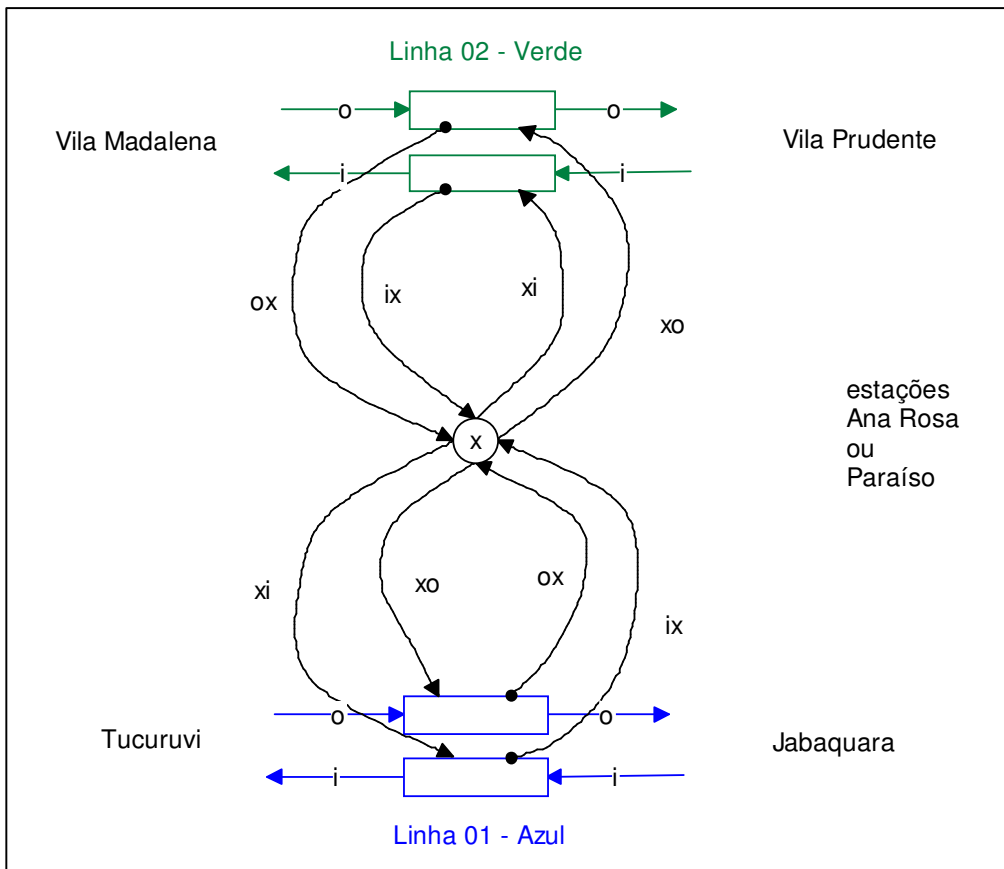


Figura 2 – Arcos de percurso de duas linhas e de transbordo entre elas – exemplo para a alternativa de rede simplificada



Para cada linha que passa por uma estação, servida por mais de uma linha, há de se criar pares de arcos de desembarque na estação para essas linhas, com as direções "O,X", "X,O", "I,X", "X,I". Isso pode ser feito com quatro *queries* similares, que variem apenas quanto às combinações de direções, mas é mais econômico fazê-lo com o auxílio de uma tabela com essas combinações.

Criar a tabela TIPO_Xfer (DIRECAOo, DIRECAOd) e instanciar com:

DIRECAOo	DIRECAOd
O	X
X	O
I	X
X	I

Em seguida, encontrar na tabela **MLS_od07** as estações com mais de uma linha para fazer a simples combinação destas com a tabela **TIPO_Xfer**:

Primeiro, a *query* **MaisDeUmaLinhaod07** para gerar a lista de estações com mais de uma linha:

```
select CODE
from MLS_od07
group by CODE
having Count(*)>1 ;
```

ou

Por meio da tabela do Modo Design:

Campo:	CODE	Contar(*)
Tabela:	MLS_od07	
Total:	Agrupar	Expressão
Mostrar:	[v]	[]
Critérios:		>1

Depois, a *query* **cria_ARCOS_xfer_od07** que faz o produto cartesiano de TIPO_XFER com o cruzamento da **MLS_od07** com **MaisDeUmaLinhaod07**, para gerar os arcos de transbordo de cada linha destas estações, com a adição dos resultados na tabela **ARCOS_od07**:

```
insert into ( C_Xo, C_Yo, C_Xd, C_Yd, CODEo, CODEd, NOMEo, NOMEd, NAMEo,
             NAMEd, LINEo, LINEd, SEQo, SEQd, DIRECAOo, DIRECAOd )
select MLS_od07.C_X, MLS_od07.C_Y, MLS_od07.C_X, MLS_od07.C_Y,
       MLS_od07.CODE, MLS_od07.CODE, MLS_od07.NOME, MLS_od07.NOME,
       MLS_od07.NAME, MLS_od07.NAME, MLS_od07.LINE, MLS_od07.LINE,
       0 as Expr1, 0 as Expr2,
       Tipo_Xfer.DIRECAOo, Tipo_Xfer.DIRECAOd
from Tipo_Xfer,
     MaisDeUmaLinhaod07 inner join MLS_od07 on
     MaisDeUmaLinhaod07.CODE = MLS_od07.CODE ;
```

ou, no Modo Design, incluir as tabelas TIPO_XFER, MLS_od07 e a *query* MaisDeUmaLinhaod07, fazer o cruzamento entre MLS_od07 e MaisDeUmaLinha, pelas suas colunas CODE:

<u>MLS_od07</u>	<u>MaisDeUmaLinhaod07</u>	<u>TIPO_XFER</u>
C_X		DIRECAOo
C_Y		DIRECAOd
CODE	-- CODE	
NOME		
NAME		
LINE		
SEQUENCE		
SINCE		

Por não haver qualquer condição de cruzamento (*join*) entre a tabela **TIPO_XFER** e as demais, todas as suas linhas serão combinadas com o resultado da junção de **MLS_od07** com **MaisDeUmaLinhaod07**, ou seja, serão gerados 4 (quatro) arcos para cada linha de cada estação (servida por mais de uma linha).

Por meio da tabela do Modo Design:

Campo:	C_X	C_Y	C_X	C_Y	CODE	CODE
Tabela:	MLS_od07	MLS_od07	MLS_od07	MLS_od07	MLS_od07	MLS_od07
Acrescentar a:	C_Xo	C_Yo	C_Xd	C_Yd	CODEo	CODEd

NOME	NOME	NAME	NAME	LINE	LINE
MLS_od07	MLS_od07	MLS_od07	MLS_od07	MLS_od07	MLS_od07
NOMEo	NOMEd	NAMEo	NAMEd	LINEo	LINEd

Expr1: 0	Expr2: 0	DIRECAOo	DIRECAOd
		Tipo_Xfer	Tipo_Xfer
SEQo	SEQd	DIRECAOo	DIRECAOd

A rigor, nos arcos de transbordo com (... , DIRECAOo, DIRECAOd) = (... , "I", "X") ou (... , "O", "X"), LINEd seriam Null; e nos arcos com (... , DIRECAOo, DIRECAOd) = (... , "X", "O") ou (... , "X", "I") , LINEo seriam Null, e tal alteração não seria difícil.

Não executar esta *query* – ver o próximo item.

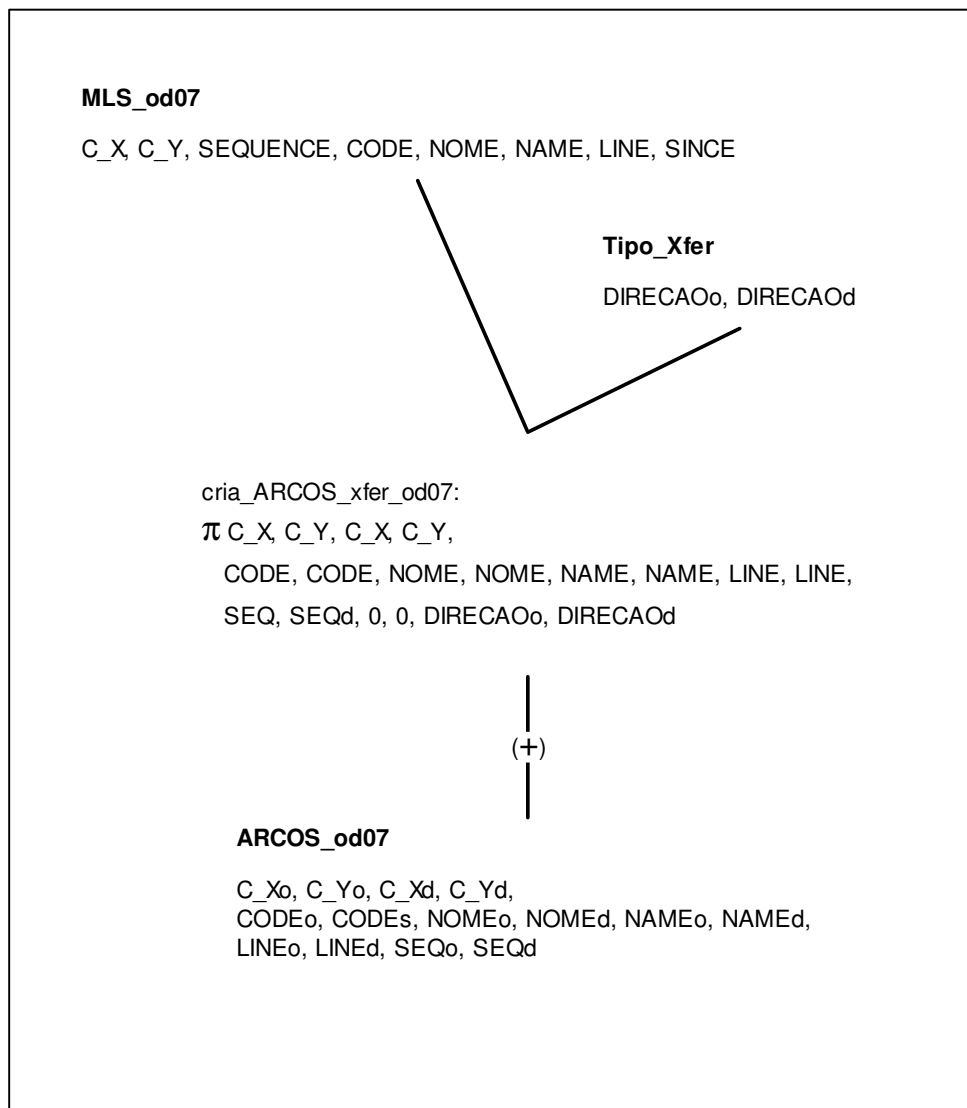
Criar arcos de transbordo nas estações operadas por uma (ou mais) linha(s)

Em situações de congestionamento, é comum os usuários embarcarem em composições na direção inversa, buscando o início da linha, quando têm dificuldade de embarcar para a direção desejada, ou quando desejam poder realizar a viagem toda sentados.

Neste caso, haveria de se criar transbordos para troca de direção de cada linha, o que indiretamente já seria possível nas estações com mais de uma linha, para o esquema alternativo proposto (veja Figura 2).

Para tanto, basta não se fazer o cruzamento da tabela **MLS_od07** com a *query* **MaisDeUmaLinhaod07**, apenas o produto cartesiano entre **MLS_od07** e **Tipo_Xfer**.

Figura 3 – Árvore de expressões para gerar os arcos de transbordo



Por outro lado, para simular o comportamento de parcela dos usuários nestas condições, haveria de se ter tempos para embarque diferentes para cada período característico do dia: antes do pico da manhã, pico da manhã, entre picos, pico da tarde e após o pico da tarde em cada linha, assim como a diferenciação nos tempos de percursos de cada linha. E ter em mente, que esses períodos variam de linha para linha.

Ainda assim, a criação de arcos de transbordo quando há apenas uma linha, cria indiretamente um nó adicional nessas estações, independente de quaisquer linhas, e que pode ser usado como ponto de partida – primeiro embarque – ou de destino – último desembarque – de viagens, respectivamente, com origem ou destino nessas estações.

Caso em que a *query* **cria_ARCOS_xfer_od07** fica como (árvore de expressões na Figura 3):

```

insert into ( C_Xo, C_Yo, C_Xd, C_Yd, CODEo, CODEd, NOMEo, NOMEd, NAMEo, NAMEd,
LINEo, LINEd, SEQo, SEQd, DIRECAOo, DIRECAOd )
select MLS_od07.C_X, MLS_od07.C_Y, MLS_od07.C_X, MLS_od07.C_Y,
       MLS_od07.CODE, MLS_od07.CODE, MLS_od07.NOME, MLS_od07.NOME,
       MLS_od07.NAME, MLS_od07.NAME, MLS_od07.LINE, MLS_od07.LINE,
       0 as Expr1, 0 as Expr2,
       Tipo_Xfer.DIRECAOo, Tipo_Xfer.DIRECAOd
from Tipo_Xfer, MLS_od07 ;

```

No Modo Design, basta remover a linha indicativa do *inner join* e, em seguida a *query* **MaisDeUmaLinhaod07**.

E, após executar essa *query*, ajustar as linhas nos arcos de transbordo, conforme citado no final do item anterior, e indicado na Figura 4:

Com a *Query* **corrige_LINEo_X_od07**

```

update ARCOS_od07
set LINEo = "---"
where DIRECAOo = "X" ;

```

ou, por meio da opção **Atualizar** do menu **Design > Tipo de Consulta** e o preenchimento da tabela de campos, do Modo Design:

Campo:	LINEo	[DIRECAOo]
Tabela:	ARCOS_od07	ARCOS_od07
Atualizar para:	"---"	
Critérios:		"X"

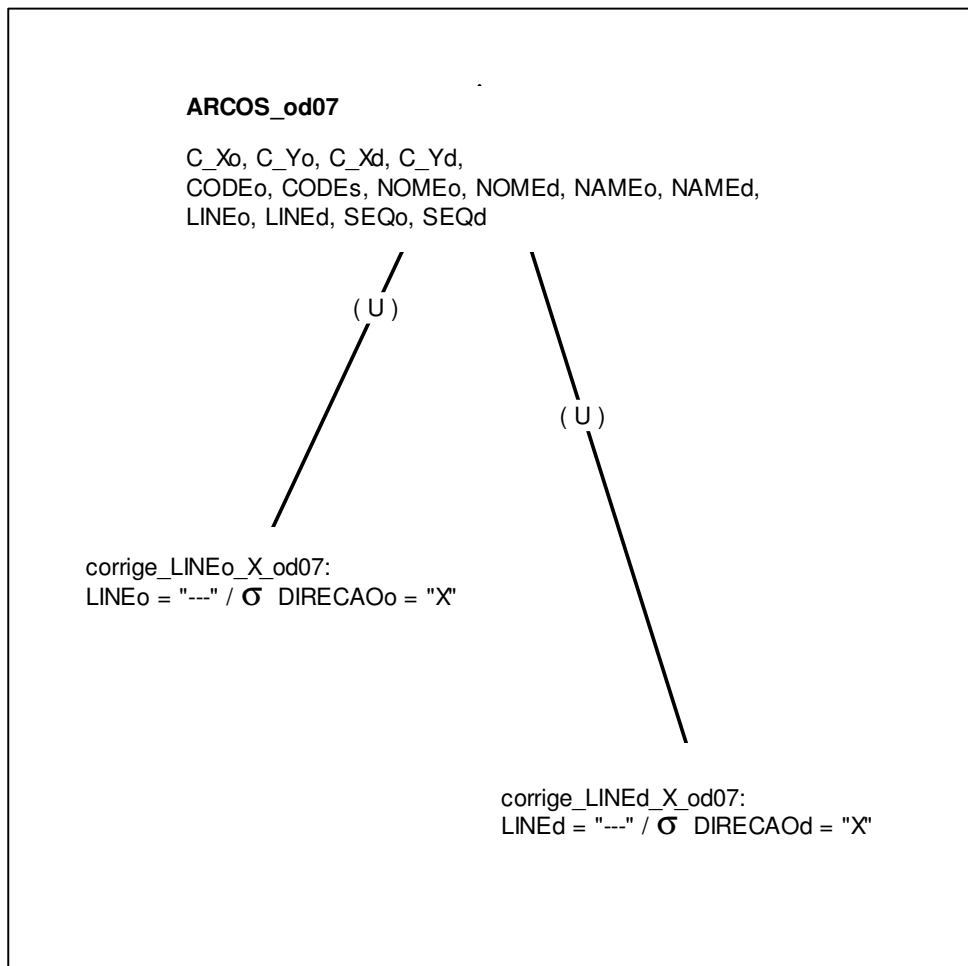
e a query corrige_LINEd_X_od07

```
update ARCOS_od07
set LINEd = "---"
where DIRECAOd = "X" ;
```

ou, por meio da opção **Atualizar** do menu **Design > Tipo de Consulta** e o preenchimento da tabela de campos, do Modo Design:

Campo:	LINEd	[DIRECAOd]
Tabela:	ARCOS_od07	ARCOS_od07
Atualizar para:	"---"	
Critérios:		"X"

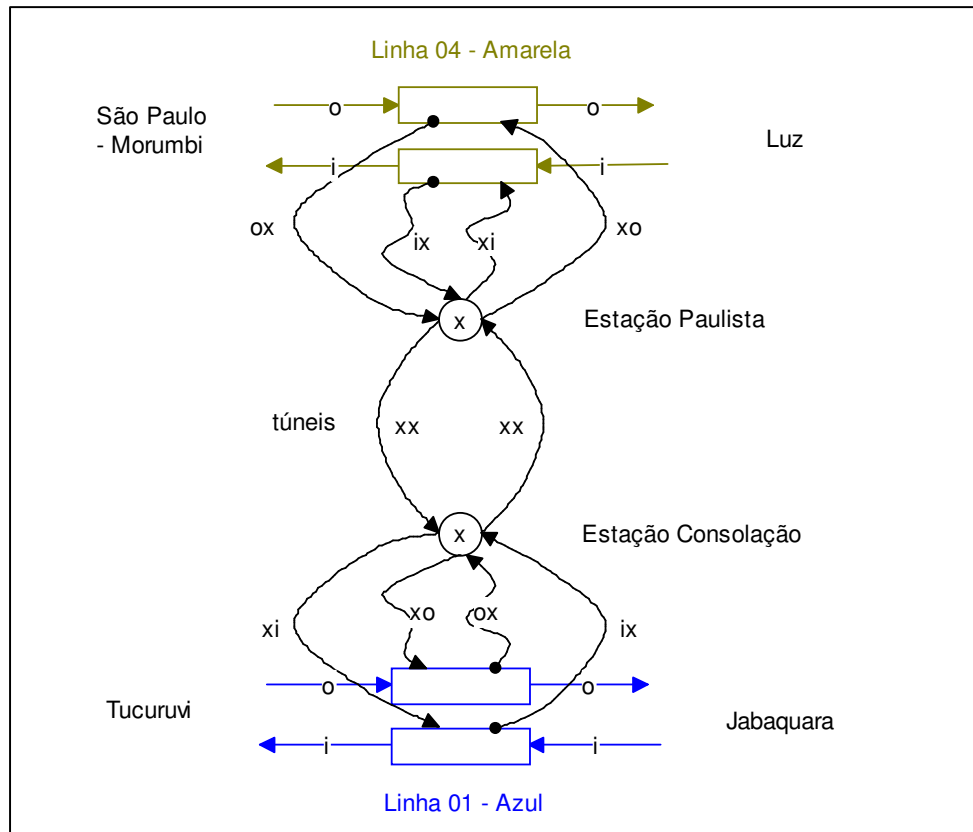
Figura 4 – Correção nos valores das colunas(LINEo e/ou LINEd, dos arcos de transbordo



Criar arcos de transbordo para as estações interligadas

Há um conjunto de transbordos que ocorre entre as estações Consolação (da Linha 02 - Verde) e Paulista (da Linha 04 – Amarela). São estações diferentes, mas com acesso subterrâneo (túneis) e integração tarifária. Neste caso, os arcos de transferência entre Linhas diferentes de estações interligadas por túneis, já para a rede simplificada, estão indicados na Figura 5.

Figura 5 – Nós fictícios nas estações Consolação e Paulista e arcos de transferência



Os arcos identificados por xx na Figura 3 teriam que ser acrescentados à tabela ARCOS_od07, com a seguinte composição:

```
330394;7394031;330631;7393668;PTA;CNS;Paulista;Consolação;Paulista;
Consolacao;04;02;0;0;X;X
```

```
330631;7393668;330394;7394031;CNS;PTA;Consolação;Paulista;Consolacao;Paulista;
02;04;0;0;X;X
```

Os demais arcos gerados pelo cruzamento das tabela MLS_od07 e Tipo_Xfer, sem a condição de haver mais de uma linha operando em cada uma destas estações, o que resultaria:

```
330394;7394031;330394;7394031;PTA;PTA;Paulista; Paulista;Paulista;
Paulista;04;04;0;0;I;X
```

330394;7394031;330394;7394031;PTA;PTA;Paulista; Paulista;Paulista;
Paulista;04;04;0;0;X;I

330394;7394031;330394;7394031;PTA;PTA;Paulista; Paulista;Paulista;
Paulista;04;04;0;0;O;X

330394;7394031;330394;7394031;PTA;PTA;Paulista; Paulista;Paulista;
Paulista;04;04;0;0;X;O

330631;7393668;330631;7393668;CNS;CNS;Consolação;Consolação;Consolacao;
Consolacao;02;02;0;0;I;X

330631;7393668;330631;7393668;CNS;CNS;Consolação;Consolação;Consolacao;
Consolacao;02;02;0;0;X;I

330631;7393668;330631;7393668;CNS;CNS;Consolação;Consolação;Consolacao;
Consolacao;02;02;0;0;O;X

330631;7393668;330631;7393668;CNS;CNS;Consolação;Consolação;Consolacao;
Consolacao;02;02;0;0; X;O

Por causa da rede em estudo, objeto deste exercício, não contemplar a Linha 4 – Amarela, ainda não inaugurada em 31.dez.2006, os arcos de transbordo para este caso não são gerados.

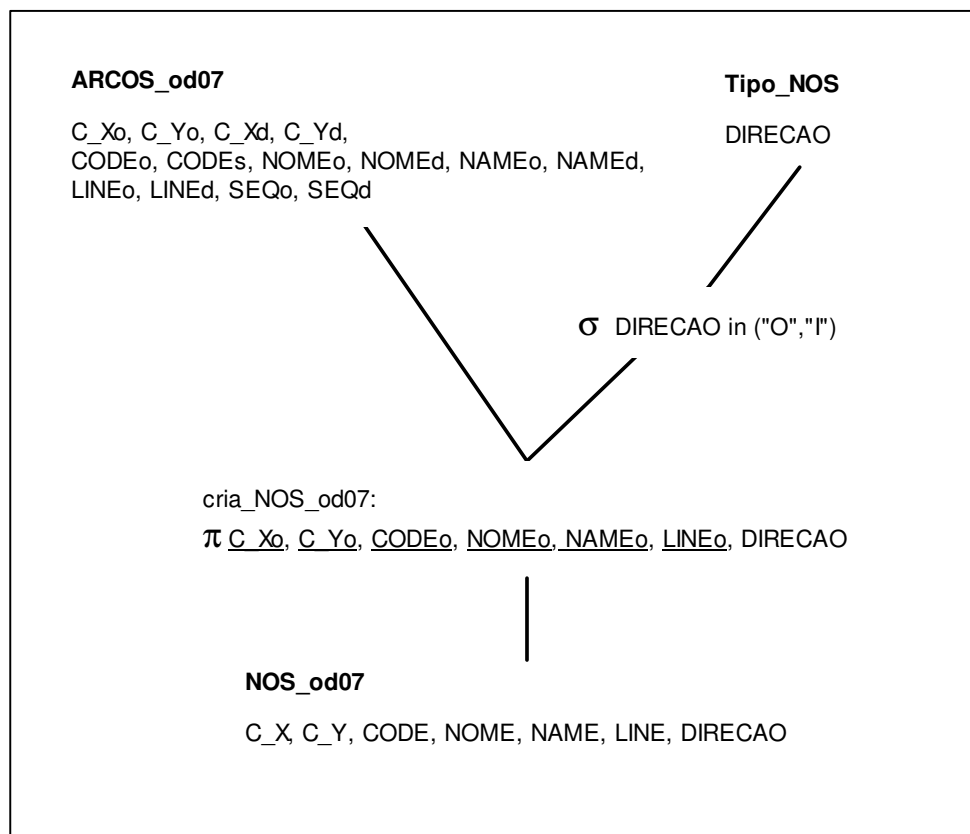
Criação dos nós da rede – a partir do conjunto completo dos arcos

Com a criação dos arcos de sentido inverso e de transbordo não se pode dizer que a tabela MLS_od07 contenha os nós da rede metro-ferroviária. Com esses arcos, foram criados, implicitamente, outros nós dentro das estações em que eles ocorrem.

Esses nós podem ser identificados, não mais pela chave candidata (LINE, CODE) de MLS_od07, mas implicitamente por (LINEo, CODEo, DIRECAOo) ou (LINEd, CODEd, DIRECAOd), com DIRECAOo/d = ("O" ou "I"), na tabela ARCOS_od07, no caso dos arcos dos trechos das linhas entre duas estações vizinhas; e por (LINEo, CODEo, 'X') para os embarques/desembarques nas estações (independente das linhas).

Há de se escrever a(s) query(ies) para, a partir da tabela ARCOS_od07, gerar a tabela de nós NOS_od07 (C_X, C_Y, LINE, CODE, NOME, NAME, DIRECAO).

Figura 6 – Árvore de expressões para a criação dos nós das estações associados aos trechos entre elas



Criar a tabela **Tipo_NOS** (DIRECAO) e instanciar com:

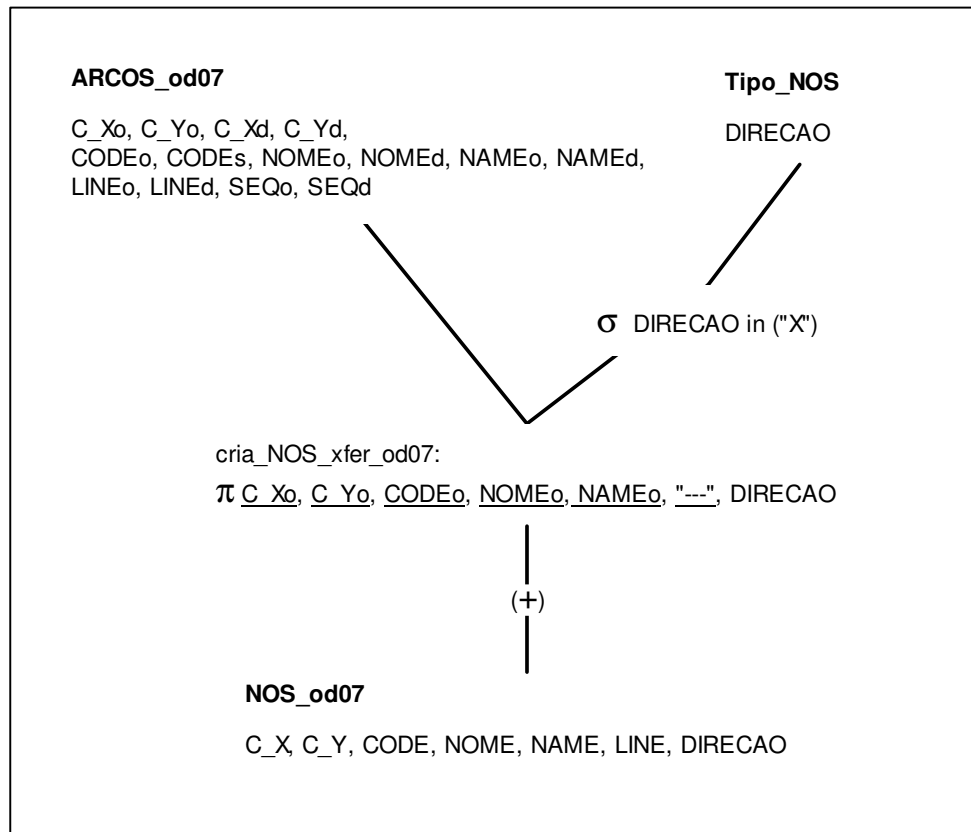
DIRECAO
O
I
X

Em seguida, criar e executar a *query cria_NOS_od07*, para criar os nós derivados das linhas, na tabela **NOS_od07**:

```
select distinct C_Xo as C_X, C_Yo as C_Y, CODEo as CODE, NOMEo as NOME,
               NAMEo as NAME, LINEo as LINE, DIRECAO
into NOS_od07
from ARCOS_od07, Tipo_NOS
where DIRECAO in ("O","I") and LINEo <> "----" ;
```

Digitada diretamente na tela Modo SQL;

Figura 7 – Árvore de expressões para a criação dos nós de transbordo de cada estação



e depois, a *query* **cria_NOS_xfer_od07**, da Figura 7, para adicionar os nós de transferência, na tabela **NOS_od07**:

```
Insert into NOS_od07 (C_X, C_Y, CODE, NOME, NAME, LINE, DIRECAO)
select distinct C_Xo, C_Yo, CODEo, NOMEo, NAMEo, "---", DIRECAO
from ARCOS_od07, Tipo_NOS
where DIRECAO in ("X") ;
```

Também digitada diretamente na tela Modo SQL.

A opção de se escrever **DIRECAO** in ("X") na cláusula **where** , ao invés de, simplesmente, **DIRECAO = "X"**, é preventiva, para a eventualidade de se criar, futuramente, outros tipos nós nas estações – caso em que bastaria acrescentar o(s) novo(s) tipo(s) à lista (atualmente com apenas o tipo X)

Nota Importante: As figuras 3, 4, 6 e 7 adicionam à documentação citada na resolução dos exercícios extraclasse da Aula 7 com MS-Access.

Até este ponto, o BD contém dez *queries*, cuja ordem de execução se requer saber, além de *verifica_data*, *gera_hml_od07*, *gera_MIF_od07* e *gera_MID_od07* (quatorze *queries* no total):

- corrige_LINEd_X_od07
- corrige_LINEo_X_od07
- cria_ARCOS_od07
- cria_ARCOS_inv_od07
- cria_MLS_od07
- cria_NOS_od07
- cria_NOS_xfer_od07
- maiores_od07
- maximo_od07
- mínimo_od07

E quatro tabelas:

- ARCOS_od07
- MetroLineStations
- MLS_od07
- NOS_od07

Mas, haverá outras.

9.2 Entradas para um programa de cálculo de caminhos mínimos

O programa executável **ShortestPath1.exe** encontra-se dentro do pacote **ShortestPath.zip**, o qual inclui exemplos de arquivos de entrada: **nodes.txt** (para os nós) e **arcs.txt** (para os arcos). Uma vez executado com sucesso, produz o arquivo **path.txt**, o qual pode ser lido em softwares de planilha eletrônica e/ou em softwares de bancos de dados.

O programa **ShortestPath1.exe** adota o algoritmo de caminhos mínimos de Dijkstra(1959).

Há também no pacote um arquivo **LeiaMe.txt** com instruções sobre os leiautes dos arquivos de entrada e saída do Programa, reproduzidas a seguir:

Arquivos de entrada:

- Arquivo dos nós (formato csv com separador ; e decimal .):

Nome : string;

ID : integer;

Raiz : integer (é Raiz de árvore de CM: 1=sim, 0=não);

Ordenado por ID

Exemplo: nodes.txt

- Arquivo dos arcos (formato csv com separador ; e decimal .):

Origem : integer (=id de um nó);

Destino : integer (=id de um nó);

Custo : real (tempo, distância, custo, custo generalizado - do arco);

Exemplo: arcs.txt

Arquivo de saída:

- Árvore caminhos mínimos (formato csv com separador ; e decimal .):

From: string (label da árvore = Nome de um dos nós Raiz de árvore de CM)

Thru: string (label da origem do trecho = Nome de um nó início de arco)

To: string (label do destino do trecho = Nome de um nó final de arco)

GCost: real (tempo, distância, custo, custo generalizado - até o final de um arco)

Exemplo: paths.txt

Observações:

- Cada par Thru-To do arquivo de caminhos mínimos tem um correspondente arco no arquivo de arcos;
- Apenas estão indicados os caminhos completos de cada nó raiz (de uma árvore CM) até os demais.

Construção dos arquivos de entrada

A estratégia sugerida para a construção dos arquivos de entrada, **nodes.txt** e **arcs.txt**, a partir das tabelas, **ARCOS_od07** e **NOS_od07**, da rede metro-ferroviária, para a data de referência de 31 de dezembro de 2006, é indicada a seguir.

- Confrontar os esquemas de **nodes.txt** e **arcs.txt** com **ARCOS_od07** e **NOS_od07**;

nodes.txt	NOS_od07
NOME	LINE & CODE & DIRECAO
ID	
Raiz	0 (inicialmente)

O par de coordenadas X e Y não tem efeito no cálculo, mas pode ser útil quando se desejar ver o resultado na forma de mapa em outros softwares;

A coluna ID é requerida pelo software de cálculo de caminhos mínimos, em seu processamento interno, mas pode ser útil para a construção de nodes e arcs, a partir de NOS_od07 e ARCOS_od07;

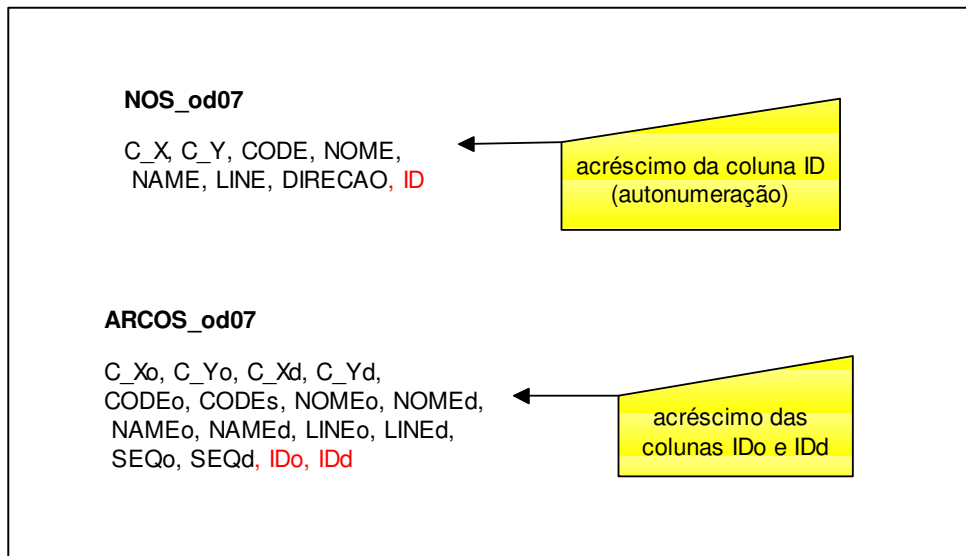
O valor de Raiz é inicialmente 0 (zero) para todas as linhas e deve ser , mais tarde, alterado para 1 (um) para cada nó cuja árvore de caminhos mínimos de deseja gerar;

É possível decompor-se NOME no arquivo de saída, **path.txt**, novamente em LINE, CODE e DIRECAO, após o processamento, pois se sabe que os tamanhos de CODE e DIRECAO são fixos.

arcs.txt	ARCOS_od07
ORIGEM: nodes.Id	
DESTINO: nodes.Id	
CUSTO	

- Escrever e executar as *queries* que permitam construir **nodes.txt** e **arcs.txt** a partir de **ARCOS_od07** e **NOS_od07**;
 - Lembrar que **NOS_od07** deve ter sido criado corretamente a partir de **ARCOS_od07**, portanto não deve haver arcos sem os correspondentes nós de saída e de entrada; e não devem haver nós que não sejam entrada ou saída de algum arco;

Figura 8 – Acrescentar colunas ID nas tabelas NOS_od07 e ARCOS_od_07



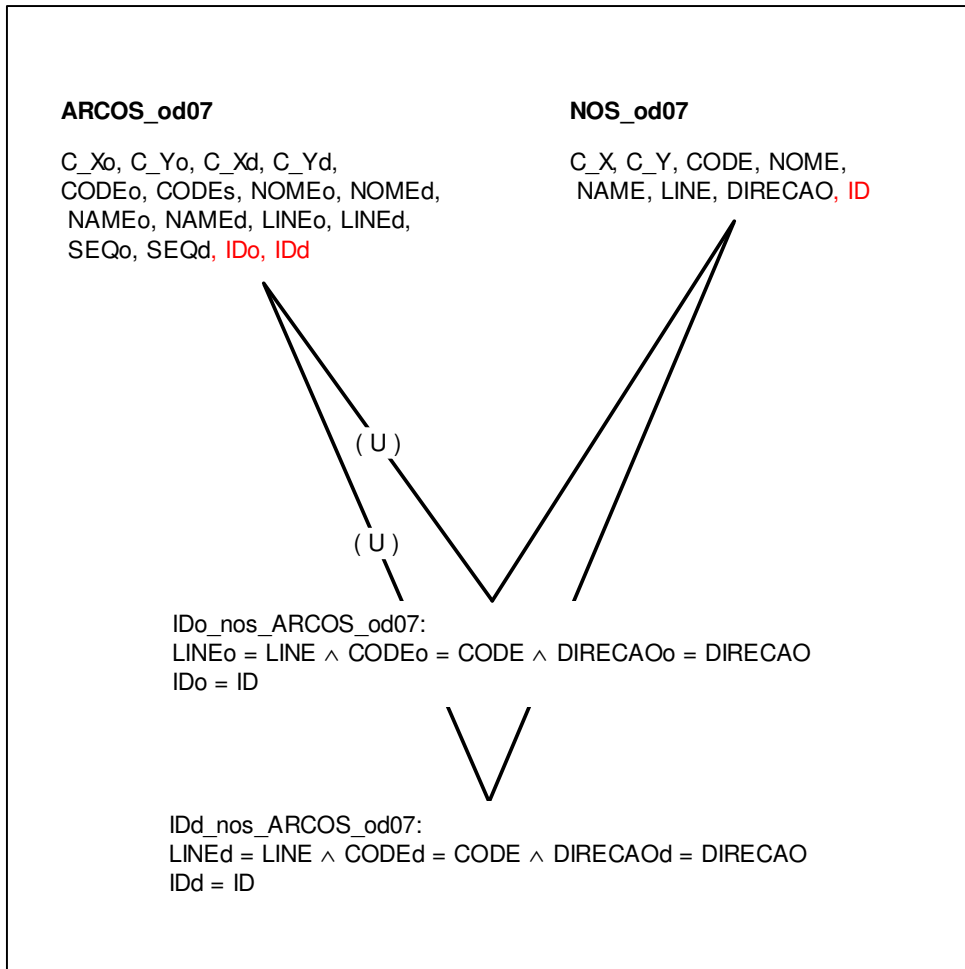
O primeiro a fazer é criar IDs para cada tupla (linha, registro, instância) da tabela **NOS_od07**. Há de alterar sua estrutura, pela criação de uma coluna, de auto-incremento, que será a chave primária dessa tabela.

- Clicar na tabela **NOS_od07** com o botão direito do mouse e selecionar a opção **Modo Design**
- Acrescentar o campo **ID** à tabela e definir seu tipo como sendo **Auto Numeração**
- Na tabela de propriedades do campo mantenha (ou atribua) os valores que seguem:
 - Tamanho do campo: **Inteiro Longo**
 - Novos valores: **Incremento (importante!)**
 - Requerido: **Sim**
 - Indexação: **Sim (Duplicação não autorizada)**
- Clicar em **Salvar** para executar a alteração na tabela.
- Fechar a janela de Modo Design da tabela.

O passo seguinte é o de criar duas colunas na tabela ARCOS_od07, uma para os IDs dos nós origem do arco e a outra para os IDs dos nós destino do arco.

- Criar as duas colunas, **IDo** e **IDd**, ambas do tipo **Número** com tamanhos **Inteiro Longo**;

Figura 9 – Atribuir valores dos IDs da tabela NOS_od07 à tabela a IDo e IDd na tabela ARCOS_od07



- Criar e executar *queries* para buscar os valores de IDo e IDd na tabela de arcos, com base nos valores de LINE, CODE e DIRECAO em cada caso: LINEo, CODEo e DIRECAOo para obter IDo; e LINEd, CODEd e DIRECAOd para obter IDd:

Com a *queries*, do tipo *update* com *join*, **IDo_nos_ARCOS_od07** para as origens:

```
update ARCOS_od07 inner join NOS_od07 on
    (ARCOS_od07.DIRECAOo = NOS_od07.DIRECAO) and
    (ARCOS_od07.LINEo = NOS_od07.LINE) and
    (ARCOS_od07.CODEo = NOS_od07.CODE)
set ARCOS_od07.IDo = [NOS_od07].[ID];
```

e **IDd_nos_ARCOS_od07** para os destinos:

```
update ARCOS_od07 inner join NOS_od07 on
    (ARCOS_od07.DIRECAOd = NOS_od07.DIRECAO) and
    (ARCOS_od07.LINEd = NOS_od07.LINE) and
    (ARCOS_od07.CODEd = NOS_od07.CODE)
set ARCOS_od07.IDd = [NOS_od07].[ID];
```

ou

No **Modo de Design** do MS-Access, para IDo:

- incluir as tabelas **ARCOS_od07** e **NOS_od07**;
- efetuar as associações correspondentes entre:

<u>ARCOS_od07</u>		<u>NOS_od07</u>
LINEo	---	LINE
CODEo	---	CODE
DIRECAOo	---	DIRECAO

- Alterar a consulta de **Selecionar** para **Atualizar - na** da barra de ferramentas, aba Design, grupo Tipo de Consulta;
- Na tabela que aparece embaixo, preencher os campos:

Campo:	IDo
Tabela:	ARCOS_od07
Atualizar para:	[NOS_od07].[ID]

Observar que o nome da tabela de nós NOS_od07 e o seu campo ID estão escritos entre colchetes []. Isso porque o MS-Access tem o mau hábito de

achar que nomes de tabelas e campos sejam texto, os coloca automaticamente entre aspas e a execução da *query* acaba falhando.

Após salvar e executar a query, fazer o mesmo para o IDd :

- Incluir as tabelas **ARCOS_od07** e **NOS_od07**;
- Efetuar as associações correspondentes entre:

<u>ARCOS_od07</u>		<u>NOS_od07</u>
LINEd	---	LINE
CODEd	---	CODE
DIRECAOd	---	DIRECAO

- Alterar a consulta de **Selecionar** para **Atualizar** - na da barra de ferramentas, aba Design, grupo Tipo de Consulta;
- Na tabela que aparece embaixo, preencher os campos:

Campo:	IDd
Tabela:	ARCOS_od07
Atualizar para:	[NOS_od07].[ID]

Há de se determinar também o **custo** para cada arco de **ARCOS_od07**, em formato numérico.

O custo é entendido no sentido mais amplo de **custo generalizado**, cuja composição pode incluir um ou mais dos seguintes:

- Tarifa paga no trecho (tempo equivalente ao custo monetário)
- Tempo de viagem do trecho (tempo)
- Penalização por desconforto no trecho (fator multiplicativo do tempo)
- Espera para embarcar (tempo)
- Penalização por desconforto, na espera para embarcar (fator multiplicativo do tempo)
- Penalização por transbordo (tempo a mais de viagem embarcado que os usuários estão dispostos a tolerar, equivalente à inconveniência de trocar de trem, ônibus, etc.)
- etc.

A questão a resolver no caso do custo composto é qual o peso se deve atribuir a cada parcela. A abordagem comum para esse tipo de problema é a de segmentar os usuários, em geral por grupo de renda e, com base na literatura, ou em pesquisa de preferência declarada (*stated preference survey*), junto a esses usuários, calibrar esses parâmetros para cada segmento.

A conversão do custo monetário em termos de tempo de viagem é diferente para cada indivíduo. Para os segmentos de maior renda espera-se um custo do tempo percebido mais alto em relação aos extratos de menor renda. Haverá aqueles para os quais o tempo importa menos (o custo monetário prevalece nas decisões), enquanto que, no outro extremo, para outros o custo monetário importa menos (o tempo é valioso).

O Banco Mundial sugere uma forma de conversão associada à renda, como a razão entre renda semanal e número de horas de trabalho na semana (GWILLIAM, 1997). Tal formulação sugere que a decisão seja individual, quando na verdade deveria considerar a renda familiar e não apenas a renda do indivíduo que realiza a viagem, por exemplo, a trabalho.

No caso ilustrativo atual, uma vez que se entra na rede a tarifa é fixa (*flat rate*), ela não irá compor o custo generalizado, o qual será dado, por exemplo, apenas pelo tempo.

Propõe-se considerado os tempos de deslocamento entre as estações e os tempos de deslocamento entre plataformas e espera para embarque nos trens. Os tempos de entrada e saída das estações, por questão de simplicidade, não serão incluídos por ocorrerem de qualquer maneira quando se entra no, e mais tarde, se deixa o, sistema metro-ferroviário.

Arcos a atribuir tempos de viagem embarcado na LINEo:

$LINEo = LINEd$, com $SEQo$ e $SEQd$ maiores do que zero (0)

Nota: $SEQo$ e $SEQd$ para arcos de desembarque e embarque ou transferência devem ser zero (0).

Arcos a atribuir tempos de deslocamento entre plataformas e de espera para embarque na LINEd:

$LINEo \leftrightarrow LINEd$, com $DIRECAOo = 'X'$ e $DIRECAOd = 'O'$

$LINEo \leftrightarrow LINEd$, com $DIRECAOo = 'X'$ e $DIRECAOd = 'I'$

Arcos a atribuir apenas o tempo de deslocamento entre plataformas, para casos como o das transferências entre as estações Consolação e Paulista:

$LINEo \leftrightarrow LINEd$, com $DIRECAOo = 'X'$ e $DIRECAOd = 'X'$

Arcos de desembarque apenas, com custo atribuído igual a zero (não pode ser null):

$LINEo \leftrightarrow LINEd$, com $DIRECAOo = 'O'$ e $DIRECAOd = 'X'$

$LINEo \leftrightarrow LINEd$, com $DIRECAOo = 'I'$ e $DIRECAOd = 'X'$

Podem-se criar tabela(s) para depois atribuir esses valores aos arcos por meio de *queries* do tipo update com join, por exemplo:

$LINHA_od07$ (LINE, TeE, IeT)

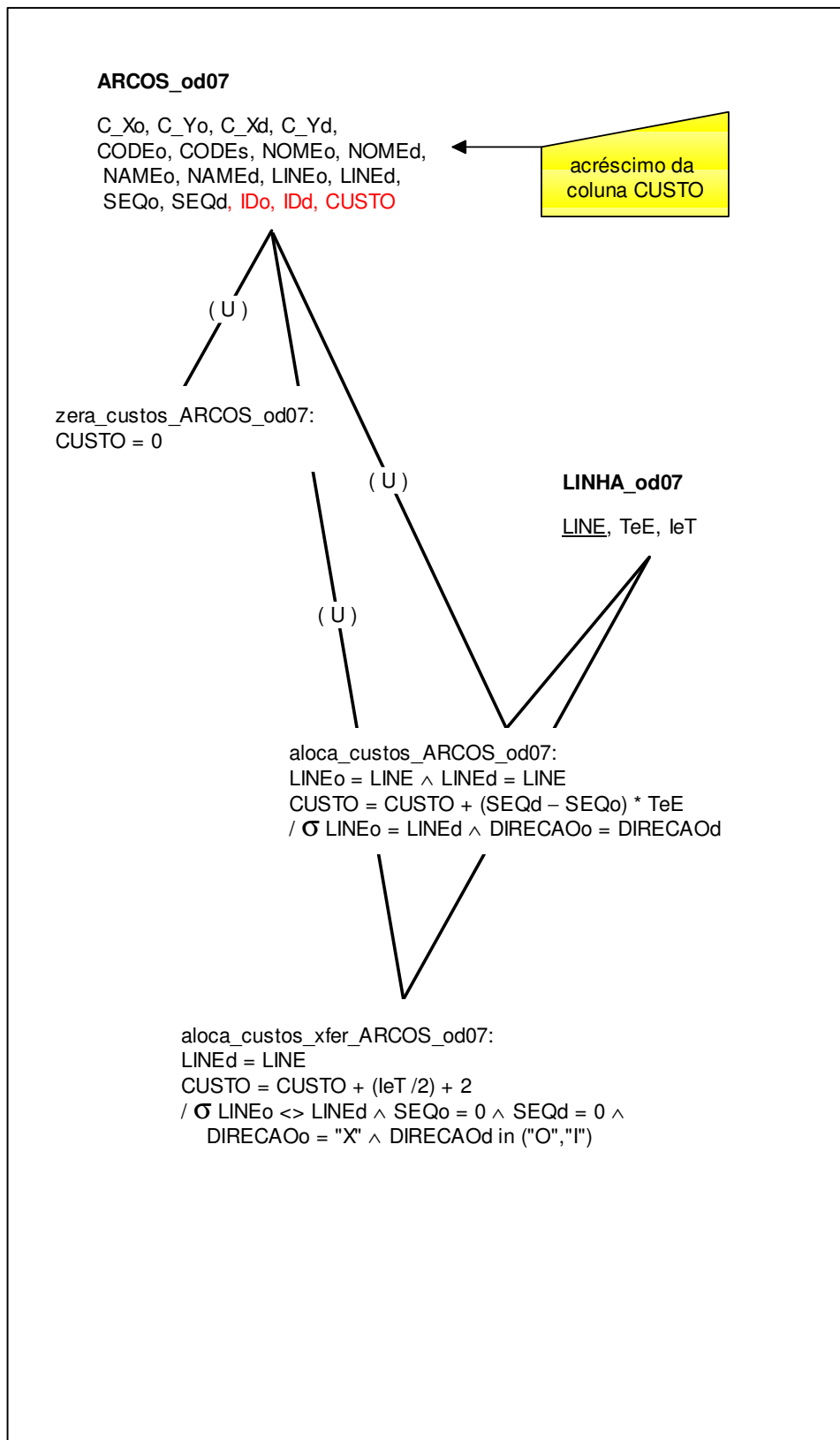
Onde TeE é o Tempo médio estações, para cada linha; e IeT é Intervalo entre trens, corresponderia ao tempo médio entre as chegadas de dois trens sucessivos (*headway*). Ambos expressos em frações decimais de minutos.

Linha		TeE ^{1,2}	IeT ²
01	Azul	1.50	2.08
02	Verde	1.93	2.13
03	Vermelha	2.04	1.98
05	Lilás	1.85	3.00
07	Rubi	4.34	6.00
07b	Rubi (extensão)	6.79	15.00
08	Diamante	2.78	6.00
09	Esmeralda	2.88	7.00
10	Turquesa	4.25	8.00
11	Coral	5.00	8.00
11b	Coral (extensão)	5.00	8.00
12	Safira	4.72	8.00

¹ . tempos médios calculados, com base em < <http://www.metro.sp.gov.br/sua-viagem/trajeto/index.aspx>> e < https://pt.wikipedia.org/wiki/Companhia_Paulista_de_Trens_Metropolitanos> (com vel.com. 38km/h)

² Intervalos CPTM nos horários de pico, em < <https://www.cptm.sp.gov.br/sua-viagem/Pages/Transfer%C3%A0ncias-e-Intervalos.aspx>>; Intervalos para linhas do Metrô, ViaQuatro e ViaMobilidade: < https://pt.wikipedia.org/wiki/Metr%C3%B4_de_S%C3%A3o_Paulo>

Figura 10 – Acréscimo da coluna CUSTO à tabela ARCOS_od07 e alocação de valores



Criar a coluna **CUSTO** na tabela **ARCOS_od07** do tipo **Numérica** e tamanho **Simples** (ou se quiser Duplo) para poder armazenar números em ponto flutuante (não restrito a valores inteiros).

Inicializar a coluna CUSTO com valor zero, com a *query* **zera_custos_ARCOS_od07**:

```
update ARCOS_od07
set CUSTO = 0 ;
```

ou, no **Modo de Design** do MS-Access:

- incluir a tabela **ARCOS_od07**
- Alterar a consulta de **Selecionar** para **Atualizar - na** da barra de ferramentas, aba Design, grupo Tipo de Consulta;
- Na tabela que aparece embaixo, preencher os campos:

Campo:	CUSTO
Tabela:	ARCOS_od07
Atualizar para:	0

O tempo alocado a um arco de viagem entre duas estações seria calculado pela *query* **aloca_custos_ARCOS_od07** como:

```
update ARCOS_od07 as A inner join LINHA_od07 as L
on (A.LINEo=L.LINE) and (A.LINEd=L.LINE)
set A.CUSTO = [A].[CUSTO]+( ([A].[SEQd]-[A].[SEQo] ) * [L].[TeE] )
where (A.LINEo = A.LINEd) and (A.DIRECAOo = A.DIRECAOd) ;
```

Ou, no **Modo de Design** do MS-Access:

- incluir as tabelas **ARCOS_od07** e **LINHAS_od07**;
- efetuar as associações correspondentes entre:

<u>ARCOS_od07</u>		<u>LINHAS_od07</u>
LINEo	---	LINE
LINEd	---	LINE

- Alterar a consulta de **Selecionar** para **Atualizar - na** da barra de ferramentas, aba Design, grupo Tipo de Consulta;

- Na tabela que aparece embaixo, preencher os campos:

Campo:	CUSTO
Tabela:	ARCOS_od07
Atualizar para:	[ARCOS_od07].[CUSTO]+((([ARCOS_od07].[SEQd]-[ARCOS_od07].[SEQo])*[LINHA_od07].[TeE])
Critério:	

LINEo	DIRECAOo
ARCOS_od07	ARCOS_od07
[ARCOS_od07].[LINEd]	[ARCOS_od07].[DIRECAOd]

E os tempos associados à espera para embarque numa linha calculado pela *query* **aloca_custos_xfer_ARCOS_od07** como:

```
update ARCOS_od07 as A inner join LINHA_od07 as L
on (A.LINEd = L.LINE)
set A.CUSTO = [A].[CUSTO] + ( [L].[IeT] / 2) + 2)
where (A.LINEo <> A.LINEd)
and (A.SEQo = 0) and (A.SEQd = 0)
and A.DIRECAOo = "X" and A.DIRECAOd in ("O","I");
```

Ou, no **Modo de Design** do MS-Access:

- incluir as tabelas **ARCOS_od07** e **LINHAS_od07**;
- efetuar as associações correspondentes entre:

```
ARCOS_od07      LINHAS_od07
LINEd           — LINE
```

- Alterar a consulta de **Selecionar** para **Atualizar** - na da barra de ferramentas, aba Design, grupo Tipo de Consulta;
- Na tabela que aparece embaixo, preencher os campos:

Campo:	CUSTO	LINEo
Tabela:	ARCOS_od07	ARCOS_od07
Atualizar para:	[ARCOS_od07].[CUSTO]+([LINHA_od07].[IeT]/2)+2	<>[ARCOS_od07].[LINEd]

Critério:		
-----------	--	--

SEQo	SEQd	DIRECAOo	DIRECAOd
ARCOS_od07	ARCOS_od07	ARCOS_od07	ARCOS_od07
0	0	"X"	("O";"I")

Observar, nas formas de montagem da *query* anterior que, a clausula and DIRECAOd in ("O","I") aparece no Modo SQL com o separador vírgula (,), e no Modo Design com o separador ponto e vírgula (;) – uma surpresa (desagradável do MS-Access)

O tempo para embarque igual ao tempo médio de espera, ou seja, metade do intervalo entre trens (*headway*). Os dois minutos adicionais para os tempos de deslocamento entre plataformas, de forma bem simplificada (só exemplo ilustrativo!).

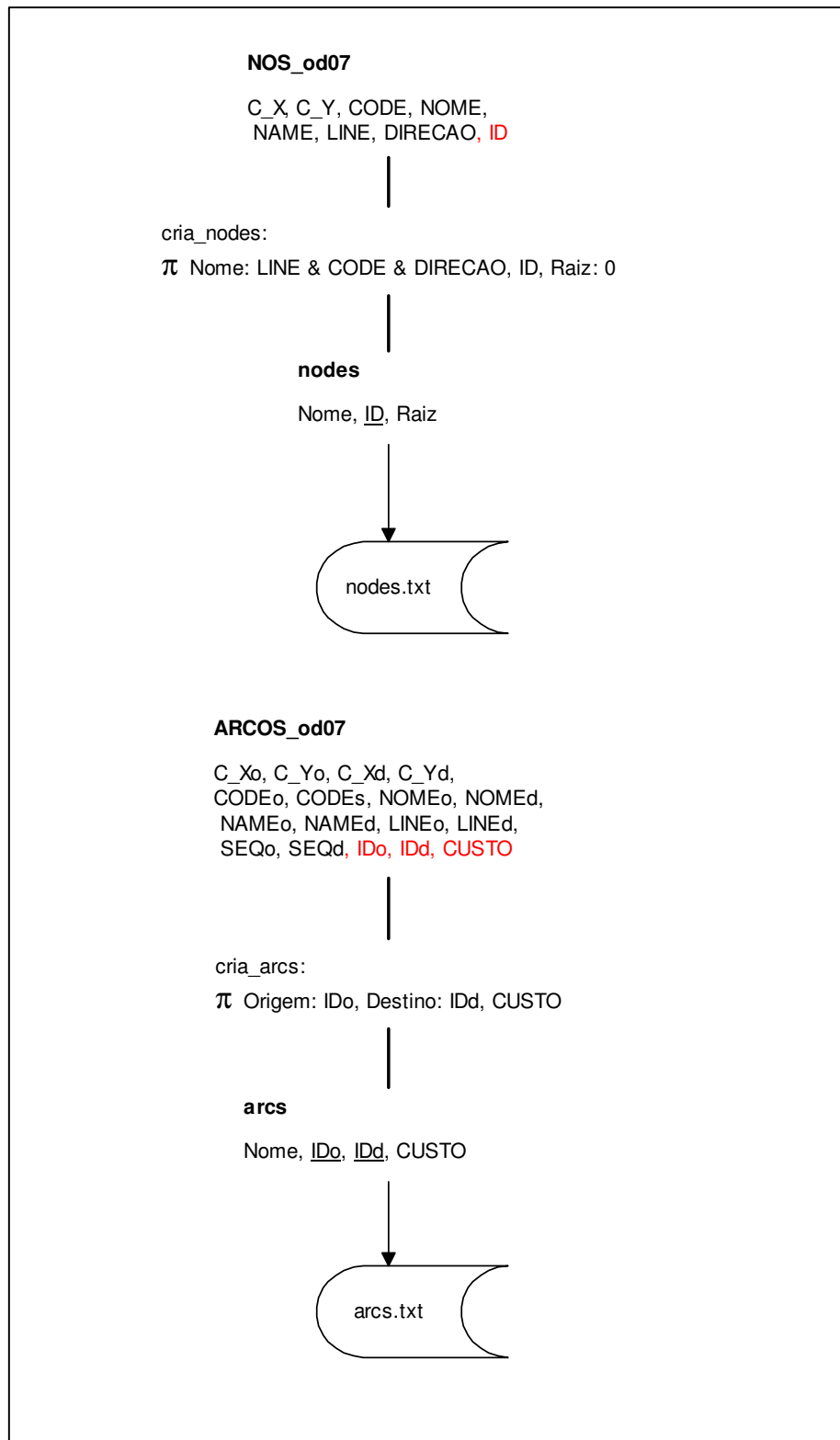
E mais dois minutos adicionais (só ilustrativo) para o caso dos túneis entre as estações, como é o caso das estações Consolação e Paulista (mas que estão fora do escopo deste exercício):

```

update ARCOS_od07 as A
set A.CUSTO = A.CUSTO + 2
where (A.LINEo <> A.LINEd)
and (A.SEQo = 0) and (A.SEQd = 0)
and A.DIRECAOo = "X" and A.DIRECAOd = "X";

```

Figura 11 – Criação das tabelas nodes e arcs, respectivamente, a partir das tabelas NOS_od07 e ARCOS_od07



Após a sequência de *queries* para atribuir custos aos arcos, pode-se criar e executar as *queries* **cria_nodes** e **cria_arcs** para criar as entradas de dados do programa de caminhos mínimos, **nodes.txt** e **arcs.txt**.

nodes.txt	NOS_od07
NOME	LINE & CODE & DIRECAO
ID	ID
RAIZ	0

```
select (LINE & CODE & DIRECAO) as NOME, ID, 0 as OD
into NODES
from NOS_od07
order by ID ;
```

e

arcs.txt	ARCOS_od07
ORIGEM	IDo
DESTINO	IDd
CUSTO	CUSTO

```
select IDo as ORIGEM, IDd as Destino, CUSTO
into ARCS
from ARCOS_od07
order by IDo, IDd ;
```

É mandatório que a lista de nós na tabela NODES seja gerada com ordenação pelo ID, pois é isso o que o programa de caminhos mínimos espera. Se for gerada outra ordem, o programa de caminhos mínimos irá embaralhar os resultados.

A tabela de nós da entrada de dados do programa de caminhos mínimos, **nodes.txt**, está com a coluna Origem/destino integralmente zerada. Se for alimentada desta forma, o programa de caminhos mínimos não irá gerar árvores de caminhos mínimos por não terem sido definidos as origens e os destinos que se deseja estudar.

Para tanto haveria de se alterar os valores dessas origens e/ou destinos para **1 (um)**, antes de se carregar os dados no programa.

Isso pode ser feito manualmente se forem poucos casos, ou por meio de *queries*, a partir de novas tabelas, de preferência.

Por exemplo:

```
Origens_Destinos_od07 (CODE, NAME, NOME)
```

Para um par, Origem-Destino, por exemplo, entre as estações Capão Redondo (CPR) da Linha Lilás (05) para a estação da Luz (LUZ) das linhas Azul (01), Rubi (07) ou Coral (11), devem ser alterados os nós de embarque/desembarque dessas estações:

```
update NOS_od_07
set OrigemDestino = 1
where right(NOME,4) in
(select CODE&"X"
from NOS_od07
where CODE in ("CPR", "LUZ")) ;
```

A *query* acima faz com que os nós "X" (embarque / desembarque) das estações de código CPR ou LUZ sejam as origens ou os destinos de viagens a serem calculadas pelo algoritmo de caminhos mínimos – caso em que serão geradas duas árvores:

- o uma, com raiz na estação CPR, e que, necessariamente terá a estação LUZ como um dos destinos (a geração desta árvore para quando se atinge a estação LUZ); e
- o a outra, com raiz na estação LUZ, e que, necessariamente terá a estação CPR como um dos destinos (a geração desta segunda árvore para em CPR).

Figura 12 – Processamento de **nodes.txt** e **arcs.txt** no programa **ShortestPaths1** e resultados em **paths.txt**

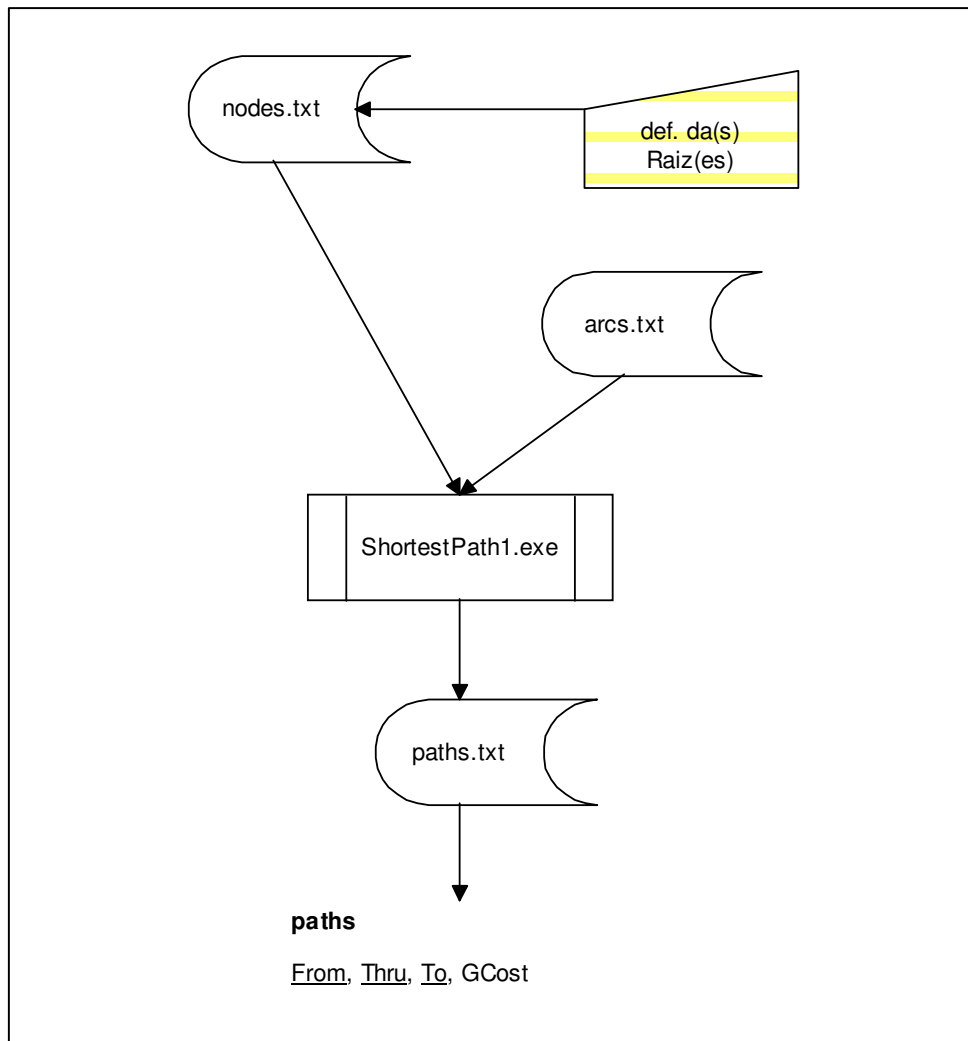
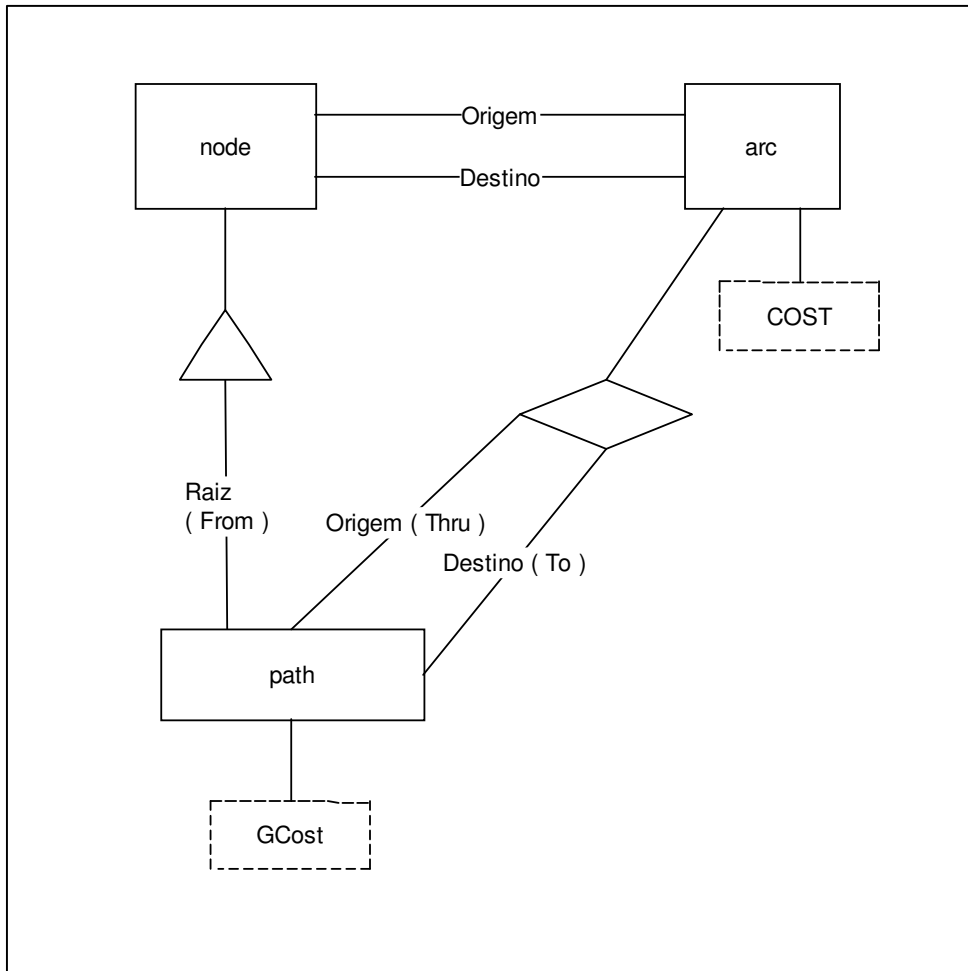


Figura 13 – Modelo ORM para a rede e caminhos mínimos



Referências:

DIJKSTRA, E. W. (1959). A note on two problems in connection with graphs (PDF). **Numerische Mathematik**. 1: 269–271. doi:10.1007/BF01386390. Disponível em: <<http://www-m3.ma.tum.de/foswiki/pub/MN0506/WebHome/dijkstra.pdf>> Acesso em: 05/06/2017.

GWILLIAM, K.M. (1997). **The value of time in economic evaluation of transport projects – lessons from recent research**. The World Bank (Transport NO. OT-5). Disponível em: <<http://siteresources.worldbank.org/INTTRANSPORT/Resources/336291-1119275973157/td-ot5.pdf>> Acesso em: 19/11/2013.