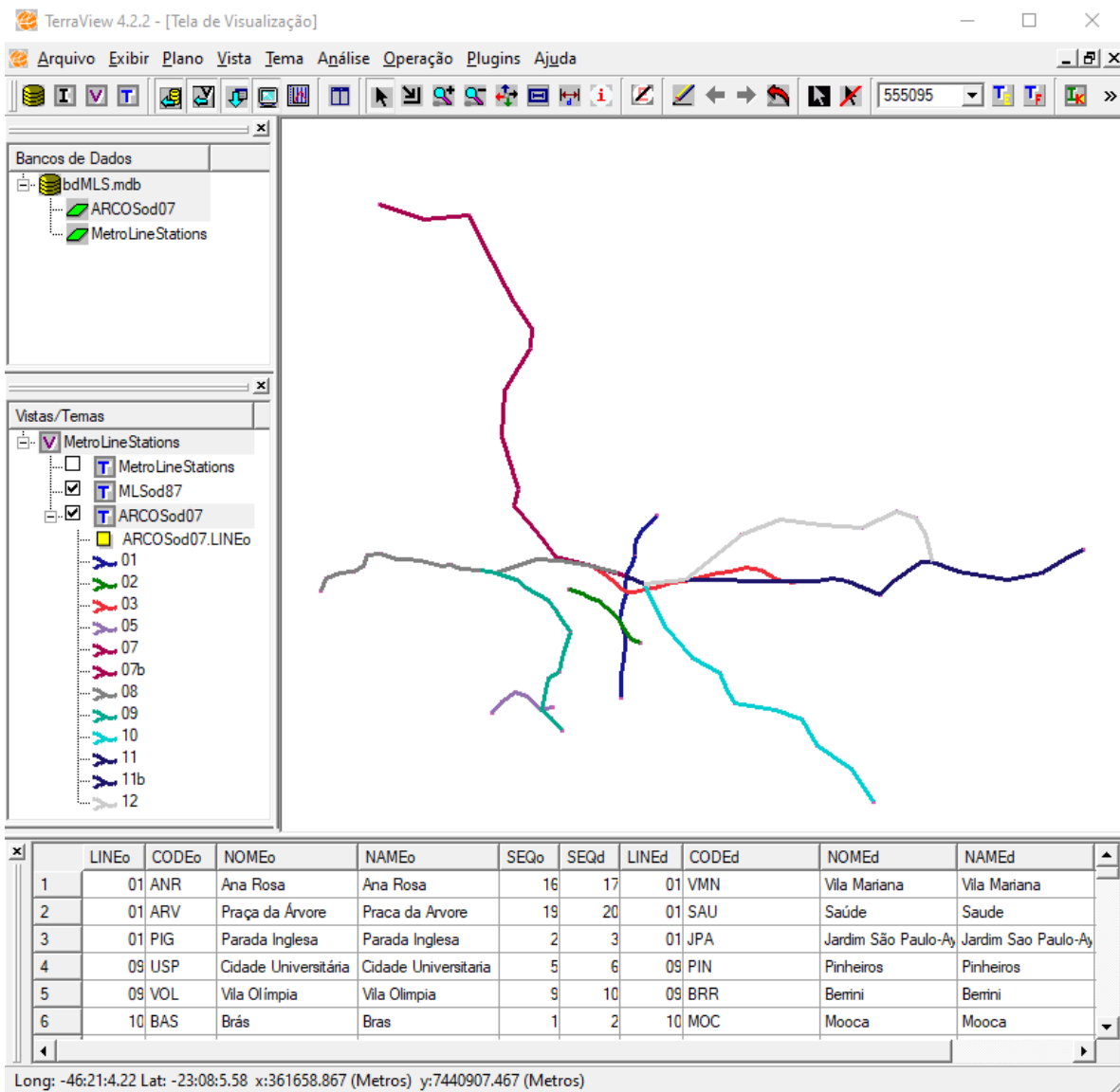


Exercício 7 – Construção de uma rede a partir da lista ordenada de seus nós

Na primeira parte do Exercício 5, desta Série, um programa em linguagem AutoLISP é executado para exportar dados para o formato MIF/MID – *MapInfo Interchange File*, a partir de objetos vetoriais (ponto e texto, linha, polilinha aberta e polilinha fechada ou polígono) em ambiente do software AutoCAD.

O formato MIF/MID pode ser lido e editado com o NotePad ou software de edição de texto, o que o torna conveniente como meio de troca de dados, tal como o XML – *Extensible Markup Language*.

Figura 1 – Linhas do sistema metro-ferroviário em 31.12.2006



Neste exercício um arquivo CSV com as coordenadas de estações da sequência das linhas do sistema metro-ferroviário da RMSP será carregado como tabela de banco de dados. Dado que em cada linha, as estações estão numeradas segundo uma das direções, serão criadas e executadas *queries* para a criação de linhas, representativas dessas sequências, não o traçado real, mas a topologia da rede metro-ferroviária. A Figura 1 ilustra exemplo dessas linhas com suas cores características.

Em se tratando de um modelo do sistema metro-ferroviário, representado como uma rede, as estações da sequência recebem a denominação de **nós**, e as ligações entre elas recebem a denominação de **arcos**. A rede em si é um **grafo**.

O arquivo desejado se encontra no sistema Tidia em: Notas de Aula e exercícios > Dados da RMSP para exercícios > Estações metro-ferroviárias (sob o cabeçalho 2016). O SRC é WGS84 / UTM 23S, como **MetroLineStations.zip**.

Carregamento dos dados em um SGBD

Após ter sido descompactado, pode-se abrir o arquivo **MetroLineStations.TXT** (na verdade trata-se de um CSV) com o NotePad ou similar. Segue um extrato do início do seu conteúdo, formatado para melhor clareza:

C_X;	C_Y;	CODE;	NOME;	NAME;	LINE;	SEQUENCE;	SINCE
336261;	7402308;	TUC;	Tucuruvi;	Tucuruvi;	01;	1;	19980429
335775;	7401578;	PIG;	Parada Inglesa;	Parada Inglesa;	01;	2;	19980429

C_X e C_Y correspondem às coordenadas em UTM da cada estação da sequência de uma linha. As colunas CODE, NOME e NAME indicam a estação. A ordem da estação, segundo uma das direções, é dada por SEQUENCE. A coluna LINE indica o número de cada linha no sistema metro-ferroviário. E a coluna SINCE indica o início de operação da linha na estação correspondente.

O arquivo de pontos pode ser carregado, tanto num BD do Terra View 4 como no QGIS 3 e, convertido para ESRI Shapefile (shp, shx, dbf, prj) ou MapInfo Interchange File (mif, mid), com o SRC definido. A conversão também para MIF/MID é útil para poder copiar os metadados do cabeçalho do MIF, editar e usar no arquivo dos arcos da rede, a construir, a partir das sequências dos nós.

A coluna LINE, das linhas, não deve ser carregada como número, até porque as extensões de algumas linhas possuem prefixos:

319113;	7429191;	BTJ;	Botujuru;	Botujuru;	07b;	2;	19080917
301170;	7394788;	SRT;	Santa Rita;	Santa Rita;	08b;	2;	20140403
362851;	7397572;	POA;	Poa;	Poa;	11b;	4;	18751106

Para cada linha que serve uma estação haverá uma instância nesse arquivo. Por exemplo, a estação da Praça da Sé aparece duas vezes:

333339;	7394613;	PSE;	Sé;	Se;	03;	6;	19790310
333272;	7394511;	PSE	Sé;	Se;	01;	11;	19780217

As coordenadas de uma estação podem variar conforme a linha, mas não muito. As datas devem ser diferentes, pois não se referem à inauguração da estação em si, mas da operação da linha nela.

Uma vez carregado como tabela num SGBD como o MS-Access ou Open Office BASE (Libre Office BASE), pode-se verificar se há coincidência nas datas para uma mesma estação, o que pode estar errado, com a *query* **verifica_data**:

```
select CODE, SINCE
from MetroLineStations
group by CODE, SINCE
having count(*) > 1 ;
```

Na *query* acima, se uma mesma estação se repete com a mesma data, seu código será exibido junto com essa data.

Nota: Os dados em **MetroLineStations** requerem verificação, mas são melhores do que os dos tutoriais do Terra View 4 do INPE.

As colunas a serem definidas como numéricas são:

C_X e C_Y	por se referirem a coordenadas;
SEQUENCE	por indicarem a ordem em cada sequência de estações.

As colunas a serem definidas como textos são:

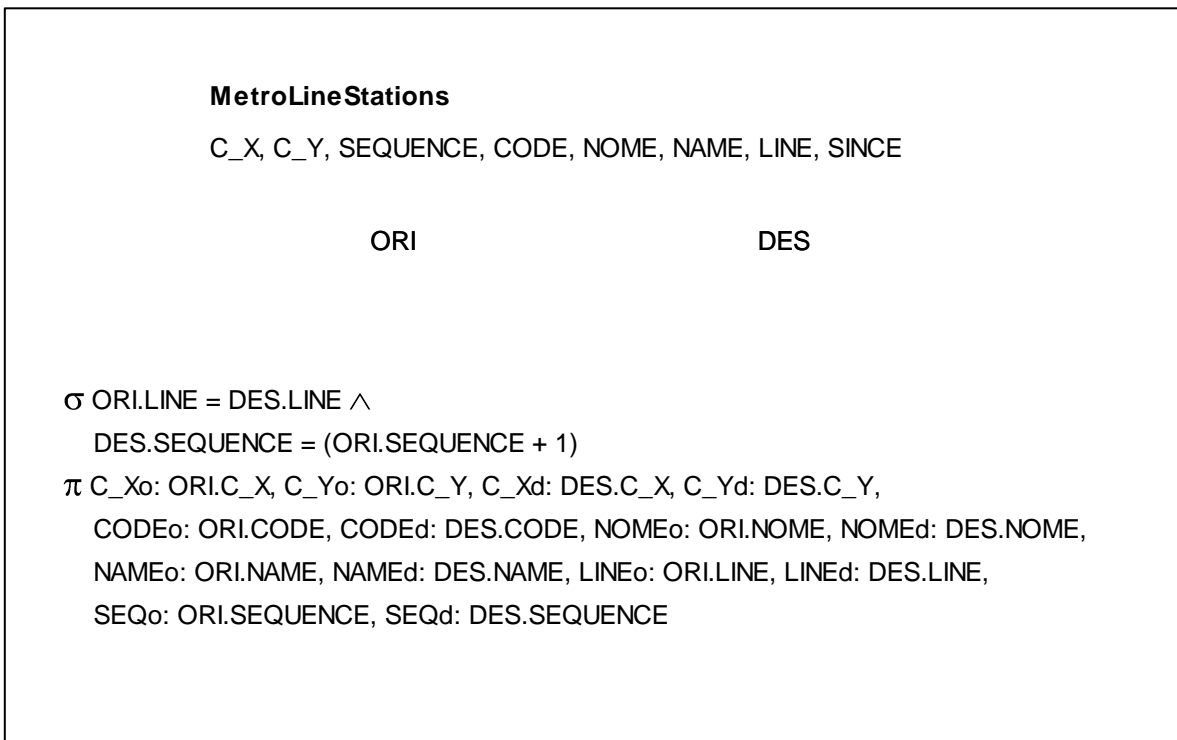
CODE, NOME e NAME	por seus conteúdos serem texto;
LINE	mesmo que não houvesse as extensões, recomenda-se tratar números que não sejam objeto de cálculos numéricos como texto;
SINCE	as datas de início de operação estão escritas num formato conveniente (YYYYMMDD) para a comparação com datas e períodos, tanto na forma numérica como na forma de texto. Ocorre que, é possível cadastrar estações ainda não inauguradas, assim como indicar datas cuja exatidão não se conhece:

323156;	7389948;	VSO;	Vila Sônia;	Vila Sonia;	04;	11;	X
304464;	7396527;	SCO;	Sagrado Coração;	Sagrado Coracao;	08;	18;	195X

Criação dos trechos entre as estações (arcos da rede) a partir das sequências de estações (os nós)

Dado que as estações de cada linha estão numeradas na ordem que os trens passam numa das direções, os trechos entre elas podem ser construídos meramente pelo cruzamento da tabela MetroLineStations com ela mesma, tendo por condição de junção que a linha seja a mesma e que a ordem de cada estação, numa das versões dessa tabela (destinos de cada trecho - DES) seja igual à ordem da estação anterior, da outra versão da tabela (origens de cada trecho - ORI), acrescida de 1 (um).

Figura 2 – Árvore de expressões para a geração dos arcos da sequência de estações de cada linha numa direção



A query abaixo produz os arcos, a partir dos nós sequenciados:

```
select ORI.C_X as C_Xo, ORI.C_Y as C_Yo,  
       DES.C_X as C_Xd, DES.C_Y as C_Yd,  
       ORI.CODE as CODEo, DES.CODE as CODEd,  
       ORI.NOME as NOMEo, DES.NOME as NOMEd,  
       ORI.NAME as NAMEo, DES.NAME as NAMEd,  
       ORI.LINE as LINEo, DES.LINE as LINEd,  
       ORI.SEQUENCE as SEQo, DES.SEQUENCE as SEQd  
from MetroLineStations as ORI inner join MetroLineStations as DES  
on (DES.SEQUENCE = ORI.SEQUENCE + 1) and (ORI.LINE = DES.LINE) ;
```

ou, alternativamente, na forma de *join* implícito:

```
select ORI.C_X as C_Xo, ORI.C_Y as C_Yo,  
       DES.C_X as C_Xd, DES.C_Y as C_Yd,  
       ORI.CODE as CODEo, DES.CODE as CODEd,  
       ORI.NOME as NOMEo, DES.NOME as NOMEd,  
       ORI.NAME as NAMEo, DES.NAME as NAMEd,  
       ORI.LINE as LINEo, DES.LINE as LINEd,  
       ORI.SEQUENCE as SEQo, DES.SEQUENCE as SEQd  
from MetroLineStations as ORI , MetroLineStations as DES  
where (DES.SEQUENCE = ORI.SEQUENCE + 1) and (ORI.LINE = DES.LINE) ;
```

Esta *query* aparentemente* resolve o problema da criação de arcos e bastariam duas *queries* subsequentes a ela, para, a partir dela, gerar os arquivos no formato MIF e MID desejados.

*Aparentemente, porque pode haver estações no arquivo MetroLineStations que ainda não foram inauguradas e não se deseja representá-las no mapa. Ou porque a rede está sendo construída para certa data, por exemplo, logo antes de uma pesquisa Origem-Destino, como a que a Companhia do Metropolitano de São Paulo – Metrô faz a cada 10 anos, desde 1967.

A filtragem das estações em MetroLineSequence para uma data, por exemplo, antes do ano da OD-2007, pode ser feita, por exemplo, pela *query* **cria_MLS_od07**, no caso do MS-Access:

```
select MetroLineStations.*  
into MLS_od07  
from MetroLineStations  
where MetroLineStations.SINCE<="20061231" ;
```

ou, no caso do Open Office BASE:

```
select MetroLineStations.*  
into cached MLS_od07  
from MetroLineStations  
where MetroLineStations.SINCE<="20061231" ;
```

Dos 176 registros de MetroLineStations, 26 foram descartados; restaram 150, quando da elaboração deste exercício.

A *query* anterior poderia se aplicada à nova tabela, **MLS_od07**, não fosse a questão da quebra da sequência das estações. Se as estações não incluídas na nova tabela estiverem nas pontas das linhas, não haveria problema, mas há casos de estações ao longo de uma linha que são inauguradas após os trens passarem por elas.

Há de se criar *queries* que resolvam o problema da quebra de sequência, que permitam formar trechos “pulando” as estações não operacionais na data indicada.

Uma solução (indicada no diagrama da Figura 3) é a de, inicialmente, determinar, para cada estação operacional de uma linha, qual a seguinte, ou seja, dentre aquelas cuja numeração maior que a dela, na sequência, qual a próxima - a de menor valor. Por exemplo, a *query* intitulada **maiores_od07**:

```
select ORI.C_X as C_Xo, ORI.C_Y as C_Yo,  
       ORI.CODE as CODEo, ORI.NOME as NOMEo,  
       ORI.NAME as NAMEo, ORI.LINE as LINEo,  
       ORI.SEQUENCE as SEQo, Maiores.SEQUENCE as SEQm  
from MLS_od07 as ORI inner join MLS_od07 as Maiores  
on ORI.LINE = Maiores.LINE  
where ORI.SEQUENCE < Maiores.SEQUENCE ;
```

ou, alternativamente, na forma de *join* implícito:

```
select ORI.C_X as C_Xo, ORI.C_Y as C_Yo,  
       ORI.CODE as CODEo, ORI.NOME as NOMEo,  
       ORI.NAME as NAMEo, ORI.LINE as LINEo,  
       ORI.SEQUENCE as SEQo, Maiores.SEQUENCE as SEQm  
from MLS_od07 as ORI , MLS_od07 as Maiores  
where ORI.LINE = Maiores.LINE  
and ORI.SEQUENCE < Maiores.SEQUENCE ;
```

Uma desvantagem desta estratégia é a criação de um grande número de combinações que serão mais tarde descartadas, mas que consomem tempo de processamento e memória, o que também resulta em maior tempo de processamento. Felizmente, neste caso, o grande número de combinações geradas (1026 combinações para 150 nós) ficou dentro do manejável pelo MS-Access 2007 (de apenas 32 bits).

A primeira estação operacional na ordem sequencial de cada linha será combinada com todas as outras. A seguinte será combinada com todas as outras, exceto a anterior. A penúltima operacional da linha será apenas combinada com a última.

A redução dessas combinações, à apenas a estação operacional seguinte, pode ser feita, a partir da *query* anterior (intitulada maiores), por meio da *query* intitulada **ori_seg_od07**:

```
select maiores_od07.C_Xo, maiores_od07.C_Yo, maiores_od07.CODEo,  
       maiores_od07.NOMEo, maiores_od07.NAMEo, maiores_od07.LINEo,  
       maiores_od07.SEQo, min(maiores_od07.SEQm) as Seguinte  
from maiores_od07  
group by maiores_od07.C_Xo, maiores_od07.C_Yo, maiores_od07.CODEo, maiores_od07.NOMEo,  
         maiores_od07.NAMEo, maiores_od07.LINEo, maiores_od07.SEQo ;
```

Trata-se de uma *query* de agregação, similar a outras de exemplos anteriores, inclusive da soma de leitos dos hospitais para cada Distrito, mas que neste caso busca o mínimo valor de **SEQm** para

cada estação de cada linha, ou seja, o número de sequência da estação imediatamente seguinte, e o atribui a **Seguinte**.

Para montar os arcos há de se combinar a *query* acima (intitulada **ori_seg_od07**) com a tabela resultante da filtragem pela data (intitulada **MLS_od07**), de modo a formarem, respectivamente, as estações inicial e final de cada trecho, em cada linha. Por exemplo, na *query* **criaARCOS_od07** para gerar a tabela **ARCOS_od07**, no caso do MS-Access:

```
select ori_seg_od07.C_Xo, ori_seg_od07.C_Yo,
       DES.C_X as C_Xd, DES.C_Y as C_Yd,
       ori_seg_od07.CODEo, DES.CODE as CODEd,
       ori_seg_od07.NOMEo, DES.NOME as NOMEd,
       ori_seg_od07.NAMEo, DES.NAME as NAMEd,
       ori_seg_od07.LINEo, DES.LINE as LINEd,
       ori_seg_od07.SEQo, DES.SEQUENCE as SEQd
into ARCOS_od07
from ori_seg_od07 inner join MLS_od07 as DES
on (ori_seg_od07.Seguinte = DES.SEQUENCE) and
   (ori_seg_od07.LINEo = DES.LINE) ;
```

ou, alternativamente, na forma de *join* implícito, no caso do Open Office BASE (também com a adição do termo “cached” na clausula “into”):

```
select ori_seg_od07.C_Xo, ori_seg_od07.C_Yo,
       DES.C_X as C_Xd, DES.C_Y as C_Yd,
       ori_seg_od07.CODEo, DES.CODE as CODEd,
       ori_seg_od07.NOMEo, DES.NOME as NOMEd,
       ori_seg_od07.NAMEo, DES.NAME as NAMEd,
       ori_seg_od07.LINEo, DES.LINE as LINEd,
       ori_seg_od07.SEQo, DES.SEQUENCE as SEQd
into cached ARCOS_od07
from ori_seg_od07 , MLS_od07 as DES
where (ori_seg_od07.Seguinte = DES.SEQUENCE) and
      (ori_seg_od07.LINEo = DES.LINE) ;
```

Nota 1: esta query apenas funciona, se feita pelo menu **Ferramentas > SQL**. Isto porque o BASE não é propriamente um SGBD (*DBMS*), mas uma interface que permite a conexão com diversos BDs (*DBs*);

Nota 2: após a criação/deleção de uma tabela no BASE, deve-se realizar um *refresh* para que ela seja exibida na lista de tabelas – o BASE acessa o BD ao qual está conectado e atualiza sua lista de tabelas – no menu **Exibir > Atualizar Tabelas**;

Obs: Uma chave candidata (a chave primária) para a nova tabela **ARCOS_od07** é a formada pela composição (LINEo, CODEo, CODEd). Como LINEd = LINEo basta uma dessas colunas na composição.

A rede existente antes da Pesquisa OD 2007 do Metrô está definida no banco de dados pelos nós, a tabela **MLS_od07**; e pelos arcos, a tabela **ARCO_Sod07**. A partir delas se podem gerar outros formatos, conforme a necessidade. Por exemplo:

- Tabela em HTML parte de uma página web;
- Arquivos MIF / MID para gerar o mapa correspondente;

Gerar uma tabela em HTML a partir da tabela **ARCOS_od07**

Com a *query* **gera_HTML_od07** no caso do MS-Access:

```
select "<tr>" &
      "<td>" & ARCOS_od07.LINEo & "</td>" &
      "<td>" & ARCOS_od07.NAMEo & "</td>" &
      "<td>" & ARCOS_od07.SEQo & "</td>" &
      "<td>" & ARCOS_od07.SEQd & "</td>" &
      "<td>" & ARCOS_od07.LINEd & "</td>" &
      "<td>" & ARCOS_od07.NAMEd & "</td>" &
      "</tr>" as table_row
from ARCOS_od07
order by ARCOS_od07.LINEo, ARCOS_od07.SEQo ;
```

ou, no caso do Open Office BASE, substituindo **&** (*ampersand* ou “e comercial”) por dois | (*pipe* ou barra vertical – no teclado a tecla à esquerda da tecla Z):

```
select '<tr>' ||
      '<td>' || ARCOS_od07.LINEo || '</td>' ||
      '<td>' || ARCOS_od07.NAMEo || '</td>' ||
      '<td>' || ARCOS_od07.SEQo || '</td>' ||
      '<td>' || ARCOS_od07.SEQd || '</td>' ||
      '<td>' || ARCOS_od07.LINEd || '</td>' ||
      '<td>' || ARCOS_od07.NAMEd || '</td>' ||
      '</tr>' as table_row
from ARCOS_od07
order by ARCOS_od07.LINEo, ARCOS_od07.SEQo
```

A *query* acima produz como resultado uma tabela com uma única coluna, onde cada linha corresponde a um trecho entre estações sucessivas de uma linha, até a data de referência, no caso 31 de dezembro de 2006. A sintaxe é a do elemento `<tr>` (*table row*) do HTML.

A cláusula **order by** faz com que a tabela resultante esteja ordenada segundo a linha e, em cada uma, segundo a ordem das estações.

Basta emendar essas linhas ao cabeçalho fixo, ilustrado a seguir, o qual pode ser aprimorado para se obter o efeito visual desejado:

```
<html>
<head>
</head>
<body>
<table>
<tr><th>Linha</th><th>Estação</th><th>#de</th><th>#para</th><th>Linha</th><th>Estação</th></tr>
```

Abaixo um extrato, das quatro primeiras e das quatro últimas linhas, do resultado da *query* **gera_HTML_od07**:

```
<tr><td>01</td><td>Tucuruvi</td><td>1</td><td>2</td><td>01</td><td>Parada Inglesa</td></tr>
<tr><td>01</td><td>Parada Inglesa</td><td>2</td><td>3</td><td>01</td><td>Jardim Sao Paulo-Ayrton Senna</td></tr>
<tr><td>01</td><td>Jardim Sao Paulo-Ayrton Senna</td><td>3</td><td>4</td><td>01</td><td>Santana</td></tr>
<tr><td>01</td><td>Santana</td><td>4</td><td>5</td><td>01</td><td>Carandiru</td></tr>
...
<tr><td>12</td><td>Itaim Paulista</td><td>10</td><td>12</td><td>12</td><td>Engenheiro Manoel Feio</td></tr>
<tr><td>12</td><td>Engenheiro Manoel Feio</td><td>12</td><td>13</td><td>12</td><td>Itaquaquecetuba</td></tr>
<tr><td>12</td><td>Itaquaquecetuba</td><td>13</td><td>14</td><td>12</td><td>Aracare</td></tr>
<tr><td>12</td><td>Aracare</td><td>14</td><td>15</td><td>12</td><td>Calmon Viana</td></tr>
```

Seguido de uma parte fixa que corresponde ao final da página em HTML:

```
</table>
</body>
</html>
```

No que resulta quando aberto com um navegador web (extrato da tela):

Linha	Estação	#de	#para	Linha	Estação
01	Tucuruvi	1	2	01	Parada Inglesa
01	Parada Inglesa	2	3	01	Jardim Sao Paulo-Ayrton Senna
01	Jardim Sao Paulo-Ayrton Senna	3	4	01	Santana
01	Santana	4	5	01	Carandiru
...					
12	Itaim Paulista	10	12	12	Engenheiro Manoel Feio
12	Engenheiro Manoel Feio	12	13	12	Itaquaquecetuba
12	Itaquaquecetuba	13	14	12	Aracare
12	Aracare	14	15	12	Calmon Viana

Gerar arquivos MIF / MID a partir da tabela ARCOS_od07

O principal objetivo deste exercício é gerar o mapa dos trechos (representados por linhas) entre estações sucessivas, de cada linha, por meio do par de arquivos MIF e MID. Devem ser escritas

duas *queries*: uma para as coordenadas, cujo resultado deve ser emendado ao cabeçalho fixo do arquivo MIF; e outra para os correspondentes dados tabulares do arquivo MID.

Importante: para garantir a correspondência, elemento a elemento, entre ambos os arquivos a cláusula **order by** deve ser a mesma em ambas *queries*.

Há de se construir o cabeçalho do metadados do arquivo **MIF**. Se o arquivo **MetroLineStations** foi exportado nesse formato, a partir do software SIG, pode-se aproveitar o cabeçalho desse arquivo como base para o novo a ser gerado, por exemplo:

```
Version 300
Charset "WindowsLatin1"
Delimiter ";"
CoordSys Earth Projection 8, 104, "m", -4.500000e+001, 0.000000e+000, 9.996000e-001,
5.000000e+005, 1.000000e+007 Bounds (298870.000000, 7373464.000000)
(379193.000000, 7433592.000000)
Columns 9
C_X Float
C_Y Float
CODE Char(3)
NOME Char(60)
NAME Char(60)
LINE Char(3)
SEQUENCE Integer
SINCE Char(8)
object_id Char(16)
Data
Point 336261.000000 7402308.000000
Point 335775.000000 7401578.000000
...
```

E alterá-lo (grifados em vermelho) para:

```
Version 300
Charset "WindowsLatin1"
Delimiter ";"
CoordSys Earth Projection 8, 104, "m", -4.500000e+001, 0.000000e+000, 9.996000e-001,
5.000000e+005, 1.000000e+007 Bounds (302489.00000, 7373464.000000)
(379193.00000, 7433592.000000)
Columns 10
LINEo Char(3)
CODEo Char(3)
NOMEo Char(60)
NAMEo Char(60)
SEQo Integer
SEQd Integer
LINEd Char(3)
CODEd Char(3)
NOMEd Char(60)
NAMEd Char(60)
Data
```

As coordenadas da estação inicial e da estação final de cada trecho serão gravadas, em cada linha, para serem emendadas ao cabeçalho de metadados acima, logo abaixo da linha **Data**, com os resultados da *query* **gera_MIF_od07** , no caso do MS-Access:

```
select "Line " & C_Xo & " " & C_Yo & " " & C_Xd & " " & C_Yd as linha
from ARCOS_od07
order by LINEo, CODEo ;
```

ou, no caso do Open Office BASE:

```
select 'Line ' || C_Xo || ' ' || C_Yo || ' ' || C_Xd || ' ' || C_Yd as linha
from ARCOS_od07
order by LINEo, CODEo;
```

Observar que após o termo Line há um espaço em branco, para que não resulte “colado” à coordenada X da primeira estação. Também que há necessidade de se separar as coordenadas por um espaço, entre cada par de aspas duplas (" "). Abaixo, extrato das três primeiras linhas do resultado da *query*:

```
Line 332837 7391136 333180 7390244
Line 332966 7387916 332834 7387029
Line 333087 7395335 333272 7394510
...
```

O arquivo **MID** não possui cabeçalho. Em cada linha, as colunas devem ser consistentes com a lista no cabeçalho de metadados do **MIF** correspondente. E as suas linhas na mesma ordem das linhas de suas coordenadas no MIF. Seu conteúdo é resultado direto da *query gera_MID_od07*, no caso do MS-Access:

```
select LINEo & ";" & CODEo & ";" & NOMEo & ";" & NAMEo & ";" & SEQo & ";" &  
      SEQd & ";" & LINEd & ";" & CODEd & ";" & NOMEd & ";" & NAMEd as linha  
from ARCOS_od07  
order by LINEo, CODEo ;
```

ou, no caso do Open Office BASE:

```
select LINEo || ';' || CODEo || ';' || NOMEo || ';' || NAMEo || ';' || SEQo || ';' ||  
      SEQd || ';' || LINEd || ';' || CODEd || ';' || NOMEd || ';' || NAMEd as linha  
from ARCOS_od07  
order by LINEo, CODEo;
```

Abaixo, extrato das três primeiras linhas do resultado:

```
01;ANR;Ana Rosa;Ana Rosa;16;17;01;VMN;Vila Mariana;Vila Mariana  
01;ARV;Praça da Árvore;Praca da Arvore;19;20;01;SAU;Saúde;Saude  
01;BTO;São Bento;Sao Bento;10;11;01;PSE;Sé;Se
```