

## PROJETO 2017

O projeto a ser implementado durante o ano letivo é um sistema de informação geográfica simplificado, como definido abaixo.

*A geographic information system (GIS) is a computer system for capturing, storing, checking, and displaying data related to positions on Earth's surface. GIS can show many different kinds of data on one map. This enables people to more easily see, analyze, and understand patterns and relationships.*

Fonte: <http://www.nationalgeographic.org/encyclopedia/geographic-information-system-gis/>

O sistema será implementado incrementalmente e em fases. É importante enfatizar que em cada fase o sistema evolui, isto é, novas funcionalidades são acrescentadas, requisitos de implementação podem ser mudados, porém, as funcionalidades existentes **devem continuar funcionais**.

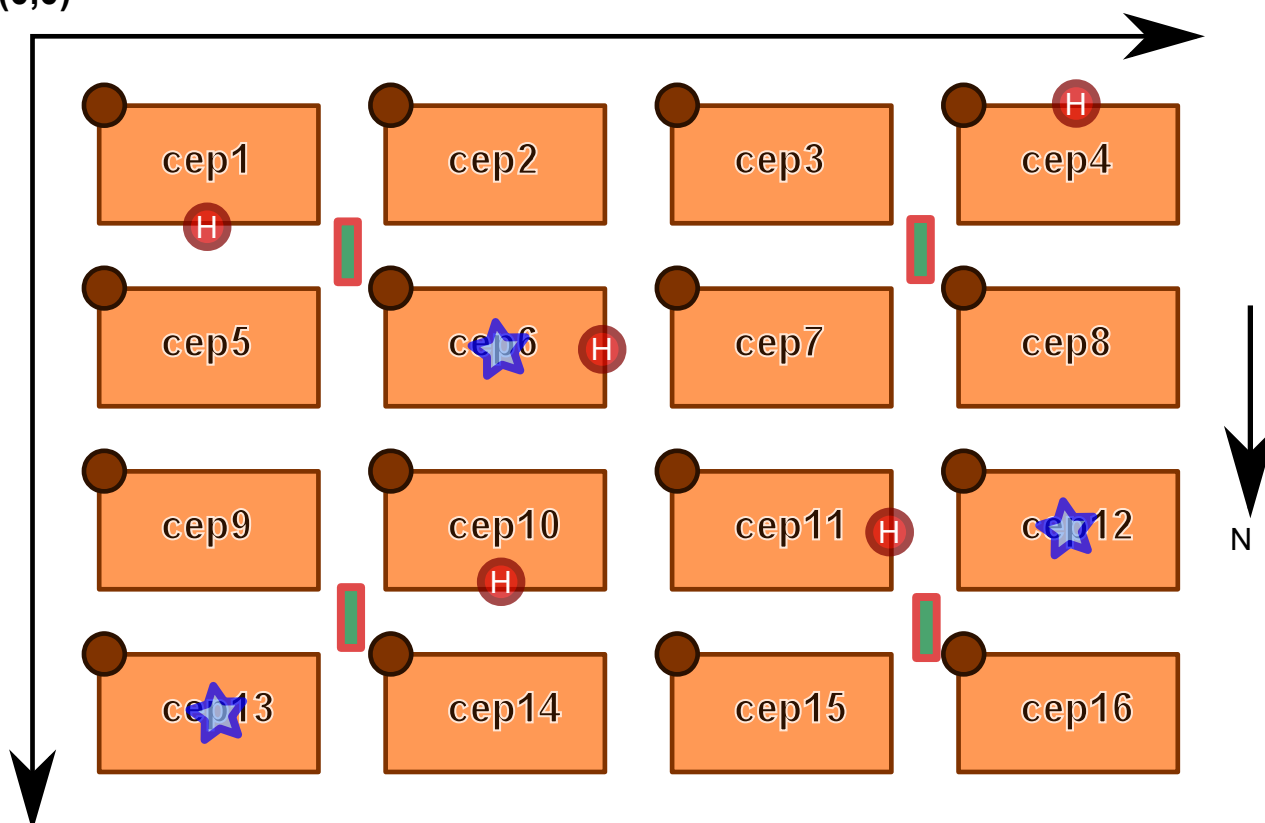
### DESCRIÇÃO

Um Sistema de Informações Geográficas (SIG), para a nossa finalidade, é um sistema que contém (não exclusivamente) dados geo-referenciados, isto é, dados com algum atributo de localização espacial (uma coordenada).

O sistema manipulará o mapa de uma cidade e algumas informações relacionadas.

O mapa de uma cidade é composto por um conjunto de retângulos que representam as quadras; e, um conjunto de equipamentos urbanos (hidrantes, semáforos, torres de celular, pontos de ônibus, etc). Cada equipamento urbano é localizado no mapa por um único ponto, conforme o exemplo abaixo.

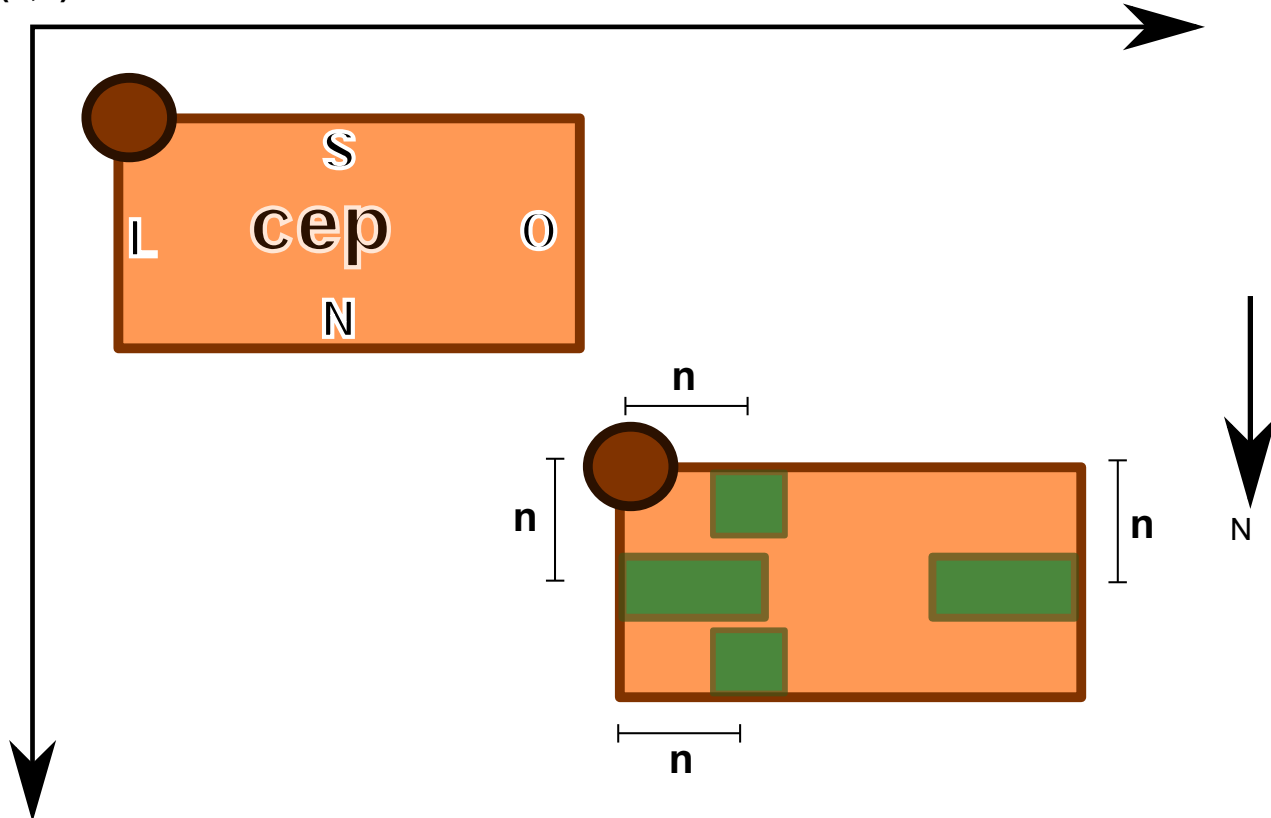
(0,0)



A cidade exemplificada acima chama-se **Bitnópolis** e possui 16 quadras. O

sistema de endereçamento de Bitnópolis é inspirado no de nossa capital federal. Cada **quadra** possui 4 **faces** (N,S,L,O) e é identificada por um **CEP** alfanumérico. O número de uma casa ou estabelecimento comercial é a **distância** da frente da casa até uma projeção do ponto de ancoragem do retângulo que representa a quadra (veja figura abaixo). Assim, um endereço é da forma CEP/Face/número, por exemplo, cep15/S/45. O ponto de ancoragem do retângulo é o canto sudeste da quadra.

(0,0)



### ENTRADA DE DADOS

A entrada de dados, via de regra, ocorrerá por meio de um ou mais arquivos. Estes arquivos estarão sob um diretório, referenciado por **BED** neste texto.<sup>1</sup>

### SAIDA DE DADOS

O dados produzidos serão mostrados na saída padrão e/ou em diversos arquivos-texto. Alguns resultados serão gráficos no formato SVG. Os arquivos de saída serão colocados sob um diretório, referenciado por **BSD** neste texto.<sup>2</sup>

### ORGANIZAÇÃO DA ENTREGA

O trabalho deve ser submetido no formato **ZIP**, cujo nome deve ser curto, mas suficiente para identificar o aluno ou a equipe.<sup>3</sup> Este arquivo deve estar organizado como descrito à frente.

<sup>1</sup> Indicado pela opção -e.

<sup>2</sup> Indicado pela opção -o.

<sup>3</sup> Por exemplo, jrsilva.zip (se aluno se chamar José Roberto da Silva), jrsilva-mrcarneiro.zip (para uma equipe com dois alunos. Evite usar maiúsculas, caracteres acentuados ou especiais).



## PROCESSO DE COMPILAÇÃO E TESTES DO TRABALHO

### Organização do ZIP a ser entregue

A organização do zip a ser entregue pelo aluno deve ser a seguinte:

#### [abreviatura-nome]

LEIA-ME.txt

\*

/src

makefile

\*.h e \*.c

*Por exemplo, jrsilva.*

*colocar matrícula e o nome do aluno. Atenção: O número da matrícula de estar no início da primeira linha do arquivo. Só colocar os números; não colocar qualquer pontuação.*

*Outros arquivos podem ser solicitados a cada fase.*

*(arquivos-fonte)*

*deve ter target para a geração do arquivo objeto de cada módulo e o target **siguel** que produzirá o executável de mesmo nome dentro do mesmo diretório **src**. Os fontes devem ser compilados com as opções **-pedantic -ansi**.*

***Atenção:** não devem existir outros arquivos além dos arquivos fontes e do makefile*

### Organização do diretório para a compilação e correção dos trabalhos (no computador do professor):

#### [HOME DIR]

\*.py

\t

\*.geo

\alunos

\abrnome

*scripts para compilar e executar*

*diretório contendo os arquivos de testes*

*arquivos de teste e, talvez, alguns outros sub-diretórios*

*(contém um diretório para cada aluno)*

*diretório pela expansão do arquivo submetido (p.e., jrsilva)*

*outros subdiretórios para os arquivos de saída informados na opção -o*

Os passos para correção serão os seguintes:

1. O arquivo .zip será descomprimido dentro do diretório alunos, conforme mostrado acima
2. O makefile provido pelo aluno será usado para compilar os módulos e produzir o executável. Os fontes serão compilados com o compilador gcc em um máquina virtual Linux. Os executáveis devem ser produzidos no mesmo diretório dos arquivos fontes O professor usará o GNU Make. Serão executadas (a partir dos scripts) o seguinte comando:
  - **make siguel**
3. O programa será executado automaticamente várias vezes: uma vez para cada arquivo de testes e o resultado produzido será inspecionado visualmente pelo

professor. Cada execução produzirá (pelo menos) um arquivo `.svg` diferente dentro do diretório informado na opção `-o`. Possivelmente serão produzidos outros arquivos `.svg` e `.txt`.

APENDICE

<https://www.gnu.org/software/make/manual/make.html>

<http://opensourceforu.com/2012/06/gnu-make-in-detail-for-beginners/>

# FASE I

## A Entrada

A entrada do algoritmo será basicamente um conjunto de retângulos e círculos dispostos numa região do plano cartesiano e, possivelmente, algumas consultas, por exemplo, que indagam se duas das formas geométricas se sobrepõem.

Considere a Ilustração 1. Cada forma geométrica é definida por uma coordenada âncora e por suas dimensões. A coordenada âncora do círculo é o seu centro e sua dimensão é definida por seu raio. A coordenada âncora do retângulo é seu canto inferior esquerdo<sup>4</sup> e suas dimensões são sua largura e sua altura. As coordenadas que posicionam as formas geométricas são valores reais e estão contidas dentro da região delimitada pelos cantos  $(0,0)$  e  $(x_{\max}, y_{\max})$ . Cada forma geométrica é indentificada por um número inteiro.<sup>5</sup>

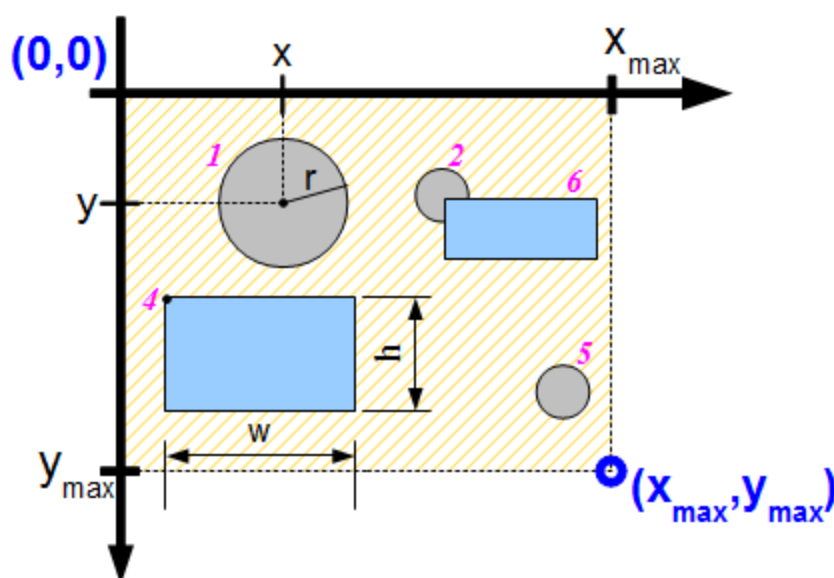


Ilustração 1

A tabela abaixo mostra o formato do arquivo de entrada, composto, basicamente, por conjunto de comandos (uma operação por linha), a saber: **c** (desenhe um círculo), **r** (desenhe um retângulo), **o** (consulta sobreposição), **i** (ponto é interno a figura?), **d** (calcule a distância entre duas figuras) e **a** (desenhe apenas as âncoras das figuras). A última linha possui a marca de final de arquivo (#).

Cada comando tem um certo número de parâmetros. Os parâmetros mais comuns são:

- $i, j, k$ : número inteiro, maior ou igual a 1. Identificador de uma forma geométrica criada pelos comandos **c** ou **r**.
- $r$ : número real. Raio do círculo.
- $x, y$ : números reais. Coordenada  $(x,y)$ .

<sup>4</sup> Note que o plano cartesiano está desenhado "de ponta-cabeça" em relação à representação usual.

<sup>5</sup>

- `cor`: string. Cor válida dentro do padrão SVG.<sup>6</sup>

| comando  | parâmetros                 | descrição   |
|----------|----------------------------|---|
| <b>c</b> | <code>i r x y cor</code>   | <i>desenhar círculo</i>   |
| <b>r</b> | <code>i w h x y cor</code> | <i>desenhar retângulo: <math>w</math> é a largura do retângulo e <math>h</math>, a altura.</i>  |
| <b>o</b> | <code>j k</code>           | <i>As formas geométricas cujos indentificadores são <math>j</math> e <math>k</math> se sobrepõem?<sup>7</sup></i>   |
| <b>i</b> | <code>j x y</code>         | <i>O ponto <math>(x,y)</math> é interno à <math>j</math>-ésima forma geométrica?<sup>8</sup></i>  |
| <b>d</b> | <code>j k</code>           | <i>Qual é a distância entre os centro de massa das formas geométricas <math>i</math> e <math>j</math>?</i>  |
| <b>a</b> | <code>sufixo cor</code>    | <i>Cria um arquivo svg, mostrando apenas os centros dos círculos e os cantos dos retângulos criados pelos comandos <code>c</code> e <code>r</code>. O nome do arquivo gerado deve ser <code>nomebase-sufixo.svg</code>. Estes pontos devem ser desenhados usando a cor <code>cor</code></i> |

A figura abaixo mostra um exemplo de um arquivo de entrada (consistente com a Figura 1). Note que a extensão do arquivo é **.geo**. As primeiras operações desenham círculos e retângulos. Na parte final do arquivo, estão colocadas duas consultas de sobreposição. A primeira pergunta se o círculo 2 e o retângulo 6 se sobrepõem (de fato, sim) e, a segunda, pergunta se os círculos 1 e 5 se sobrepõem (de fato, não). O último comando solicita que seja produzido um outro arquivo **.svg** (`a01-pl.svg`) mostrando apenas os cantos dos retângulos e os centros dos círculos.

```

c 1 50.00 50.0 30.00 grey
r 6 121.0 46.0 100.0 30.0 cyan
c 2 120.0 45.0 15.0 grey
r 4 10.0 150.0 90.0 40.0 cyan
c 5 230.0 180.0 13.0 grey
o 2 6
o 1 5
a pl red
#

```

**a01.geo**

## A Saída

O programa deverá produzir um arquivo **.svg** e um arquivo **.txt** ambos com o mesmo

<sup>6</sup> <http://www.december.com/html/spec/colorsvg.html>.

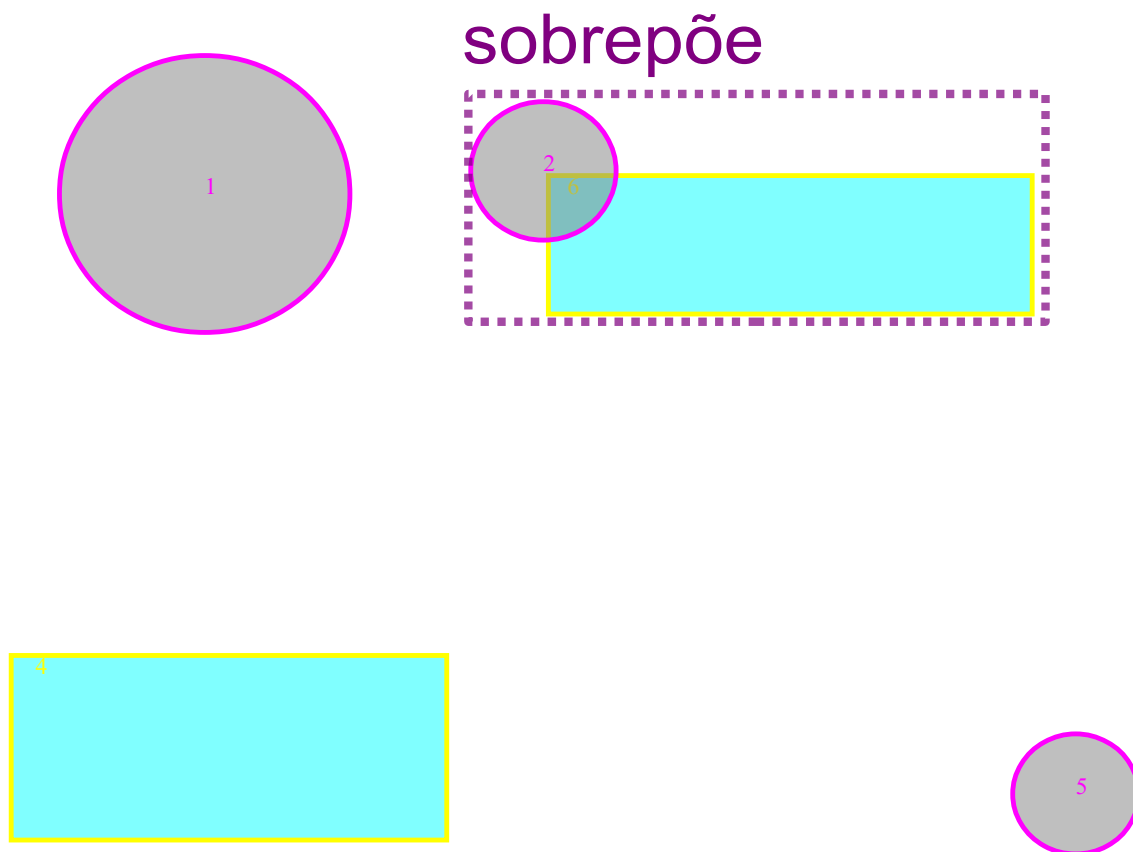
<https://www.w3.org/Graphics/SVG/IG/resources/svgprimer.html>

<sup>7</sup> A borda da figura pertence à figura. Assim, as figuras que coincidem apenas nas bordas também se sobrepõem.

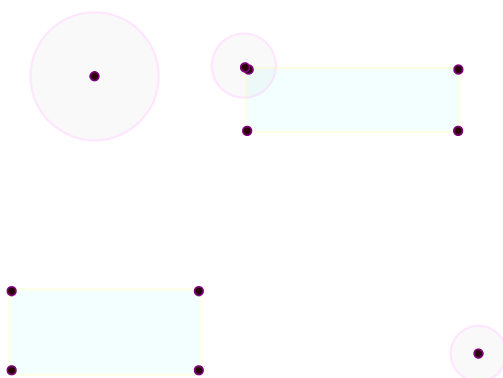
<sup>8</sup> Um ponto na borda da figura pertence à figura, **mas não é interno** à figura.

nome base do arquivo **.geo**.

O arquivo **.svg** produzido deve mostrar as formas geométricas. Além disso, para o comando **o**, caso as figuras identificadas se sobreponham, elas devem ser envolvidas por um retângulo tracejado contendo a palavra "sobreposição". Existem várias ferramentas que renderizam arquivos **.svg**. As figuras abaixo mostram um exemplo de arquivo **.svg** e sua respectiva renderização.



A operação **a** deve produzir um outro arquivo **.svg**, com nome *nomebase*-sufixo.svg, (no exemplo, *a01-pl.svg*) contendo apenas os cantos dos retângulos e centro dos círculos referentes aos comandos **c** e **r** processados até o momento, exemplificado na figura abaixo.<sup>9</sup>



O processamento de um arquivo **.geo** deve também produzir um arquivo-texto contendo o

<sup>9</sup> A figura do exemplo apresenta as "sombras" dos círculos e retângulos apenas para facilitar o entendimento. No arquivo *a01-pl.svg* são apenas produzidos os pontos escuros.



resultado das consultas *i*, *d*, *o*. Este arquivo-texto deve ser nomeado `nome-base.txt`. Neste arquivo deve ser copiado em uma linha o texto da consulta e, na linha seguinte, o seu resultado.

## O Programa

O nome do programa deve ser `t2` e aceitar dois parâmetros:

```
t2 -f arq.geo -o dir -n num
```

O primeiro parâmetro (`-f`) especifica o nome do arquivo de entrada e o segundo parâmetro (`-o`) indica o diretório onde os arquivos de saída (*arq.svg* e *arq.txt*) deve ser colocado. Note que o nome do arquivo pode ser precedido por um caminho absoluto ou relativo e *dir* é um caminho absoluto ou relativo. O parâmetro `-n` informa o número máximo de operações *r* e *c* contidas no arquivo.

A seguir, alguns exemplos de possíveis invocações de `t2`:

- `t2 -f /home/ed/testes/t001.geo -o /home/ed/alunos/aluno1/o/ -n 100`
- `t2 -f /home/ed/testes/t001.geo -n 1000 -o /home/ed/alunos/aluno1/o`
- `t2 -n 10000 -f ./testes/t001.geo -o /home/ed/alunos/aluno1/o/`
- `t2 -f ./testes/t001.geo -n 100000 -o ./alunos/aluno1/o`

## FASE II

Nesta fase serão acrescentados novos comandos ao arquivo **.geo**:

| comando                               | parâmetros       |   |
|---------------------------------------|------------------|---|
| <b>q</b>                              | x y larg alt cep | <i>Inserir uma quadra (retângulo e cep)</i>   |
| <b>h</b>                              | x y id           | <i>Inserir um hidrante</i>  |
| <b>s</b>                              | x y id           | <i>Inserir um semáforo</i>  |
| <b>t</b>                              | x y id           | <i>Inserir uma rádio-base (torre de celular)</i>  |
| <b>cq</b>                             | cfill cstrk      | <i>Cores do preenchimento e da borda das quadras (a partir deste comando)</i>           |
| <b>ch</b>                             | cfill cstrk      | <i>Cores do preenchimento e da borda dos hidrantes (a partir deste comando)</i>         |
| <b>ct</b>                             | cfill cstrk      | <i>Cores do preenchimento e da borda das torres de celular (a partir deste comando)</i> |
| <b>cs</b>                             | cfill cstrk      | <i>Cores do preenchimento e da borda dos semáforos (a partir deste comando)</i>         |
| Novos comandos do arquivo <b>.geo</b> |                  |   |

Alguns comandos de atualização e consulta podem ser colocados em um arquivo **.qry**:<sup>10</sup>

| comando   | parâmetros   |  |
|-----------|--------------|--|
| <b>dq</b> | x y larg alt | <i>remove todas quadras que estiverem inteiramente dentro do retângulo determinado pelos parâmetros do comando. Obs. Este comando, em particular, deve remover uma quadra inserida pelo comando <b>q</b> com idênticos parâmetros x, y, larg e alt. No arquivo .txt, deve apresentar os ceps das quadras removidas</i> |
| <b>dh</b> | x y larg alt | <i>Semelhante ao dq. Remove os hidrantes dentro da região. (reporta o id)</i>  |
| <b>ds</b> | x y larg alt | <i>Semelhante dq. Remove os semáforos da região.</i>   |
| <b>dt</b> | x y larg alt | <i>Semelhante dq. Remove as torres de celular dentro da região.</i>  |

<sup>10</sup> O arquivo **.qry** será informado pelo parâmetro **-q**.

| comando                  | parâmetros   |  |
|--------------------------|--------------|--|
| <b>Dq</b>                | x y raio     | <i>Remove todas as quadras que estiverem inteiramente contidas dentro do círculo de centro em (x,y) e de raio raio. Reporta no arquivo .txt o cep das quadras.</i> |
| <b>Dh</b>                | x y raio     | <i>Semelhante a Dq. Remove os hidrantes dentro da região. (Reporta o id)</i>   |
| <b>Ds</b>                | x y raio     | <i>Semelhante a Dq. Remove os semáforos dentro da região. (Reporta o id)</i>   |
| <b>Dt</b>                | x y raio     | <i>Semelhante a Dq. Remove as torres de celular dentro da região. (Reporta o id)</i>   |
| <b>crd?</b>              | ( cep   id ) | <i>Imprime no arquivo .txt as coordenadas e a espécie do equipamento urbano de um determinado cep ou com uma determinada identificação.</i>                        |
| Comandos do arquivo .gry |              |  |

## EXEMPLOS

```

cq blue black
ch red yellow
ct black red
q 37.00 15.00 89.00 40.00 cep_001-10
q 137.00 15.00 89.00 40.00 cep_001-20
q 237.00 15.00 89.00 40.00 cep_001-30
cq yellow green
q 37.00 115.00 89.00 40.00 cep_002-10
q 137.00 115.00 89.00 40.00 cep_002-20
q 237.00 115.00 89.00 40.00 cep_002-30
h 40.00 60.00 h-12
c 3 29.00 720.00 458.00 green
r 4 5.00 29.00 1049.00 479.00 green
r 5 55.00 42.00 215.00 702.00 green
c 6 51.00 139.00 635.00 chocolate

```

**a1.geo**

```

Dq 100.50 99.9 100.0
crd? cep_001-10
crd? h-12

```

**q1.gry**

## A SAIDA

A arquivo .geo (com os novos comandos) deve continuar produzindo as mesmas saídas da

fase anterior. As quadras devem ser renderizados como retângulos e pintadas como determinado pelo comando `cq`. Os outros equipamentos urbanos devem ser similarmente pintados com as cores de contorno e de preenchimento informadas pelos respectivos comandos. O arquivo `.svg` deve ser produzido após o arquivo `.qry` (se existir) tiver sido processado.

Além disso, deverá ser produzido, no diretório de saída<sup>11</sup>, um arquivo denominado `resumo.txt` com um resumo da bateria de testes.<sup>12</sup> Este arquivo deverá conter uma linha para cada um dos testes realizados. Cada linha deve ter os seguintes campos (separados por `\t`):

|   |
|---|
| <code>nome-arq.geo</code> <code>#ins</code> <code>#cpi</code> <code>#del</code> <code>#cpd</code> |
|---|

|                         |
|-------------------------|
| <code>resumo.txt</code> |
|-------------------------|

O primeiro campo é o nome do arquivo processado e os outros campos são, respectivamente, os números totais de quadras inseridas, o número total de comparações para inserir as quadras; o número total de quadras removidas e o número total de comparações para removê-las.

## A IMPLEMENTAÇÃO

O TAD Listas mostrado em sala de aula deve ser completamente implementado. As quadras e equipamentos urbanos devem ser armazenados em listas diferentes.

---

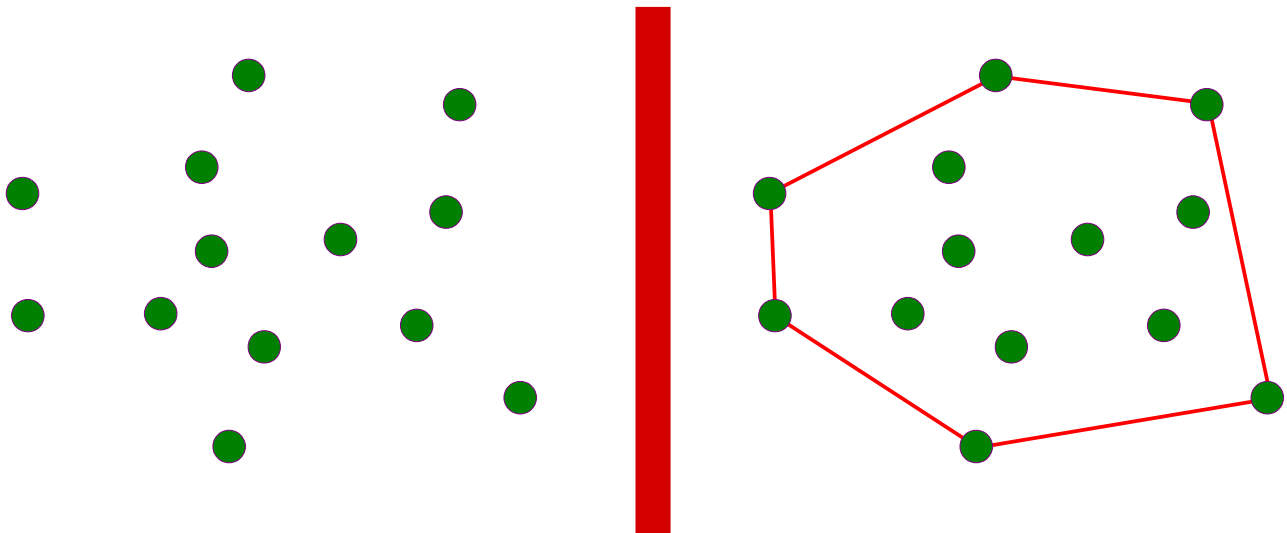
<sup>11</sup> Opção `-o`.

<sup>12</sup> O início da bateria de testes é indicado pela opção `-acc0` e a continuação indicada por `-acc`.

## FASE III

### ENVOLTÓRIA CONVEXA

A envoltória convexa de um conjunto  $Q$  de pontos num plano, denotada por  $CH(Q)$ , é o menor polígono convexo  $P$  para o qual cada ponto em  $Q$  está no contorno de  $P$  ou em seu interior.



O algoritmo para calcular a CH de um conjunto de pontos é muito simples e depende basicamente de uma ordenação particular destes pontos. Para fazer esta implementação, poderá ser utilizado algum dentre os seguintes algoritmos de ordenação: quicksort, mergesort ou heapsort.

Uma breve explicação sobre um dos algoritmos (Graham Scan) para o cálculo da envoltória convexa pode (deve) ser vista no YouTube:

<https://youtu.be/0HZaRu5IupM>

Outros materiais são facilmente encontrados usando as palavras-chave: *convex hull*, *computational geometry*, *graham scan*.

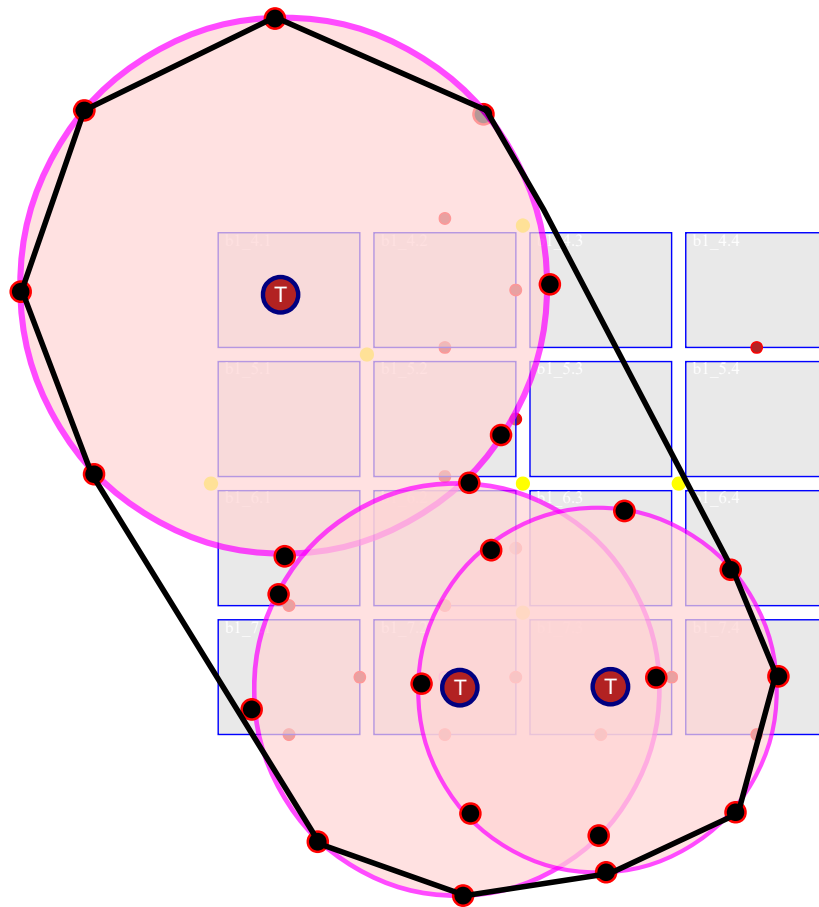
### POLIGONO DE COBERTURA DE UM CONJUNTO DE RÁDIOS-BASE

É uma aproximação da área real de cobertura de um conjunto de rádio-bases de uma região.

Note que a figura abaixo apresenta um bairro com 3 rádio-bases (círculo rotulado com T). A abrangência de cada rádio-base está indicada por uma circunferência cujo centro é a rádio-base. Se considerarmos alguns pontos nos limites de cada área de abrangência e calcularmos a envoltória convexa deste conjunto de pontos, obtemos o polígono de cobertura.<sup>13</sup> Note que o polígono de cobertura pode excluir áreas que efetivamente estão cobertas e incluir outras que não estão cobertas (por isso, é uma aproximação).

---

<sup>13</sup> Polígono de cobertura é apenas um conceito criado para este projeto.



## QUADTREE

A **quadtree** is a tree data structure in which each internal node has exactly four children. Quadrees are the two-dimensional analog of octrees and are most often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions. The data associated with a leaf cell varies by application, but the leaf cell represents a "unit of interesting spatial information".

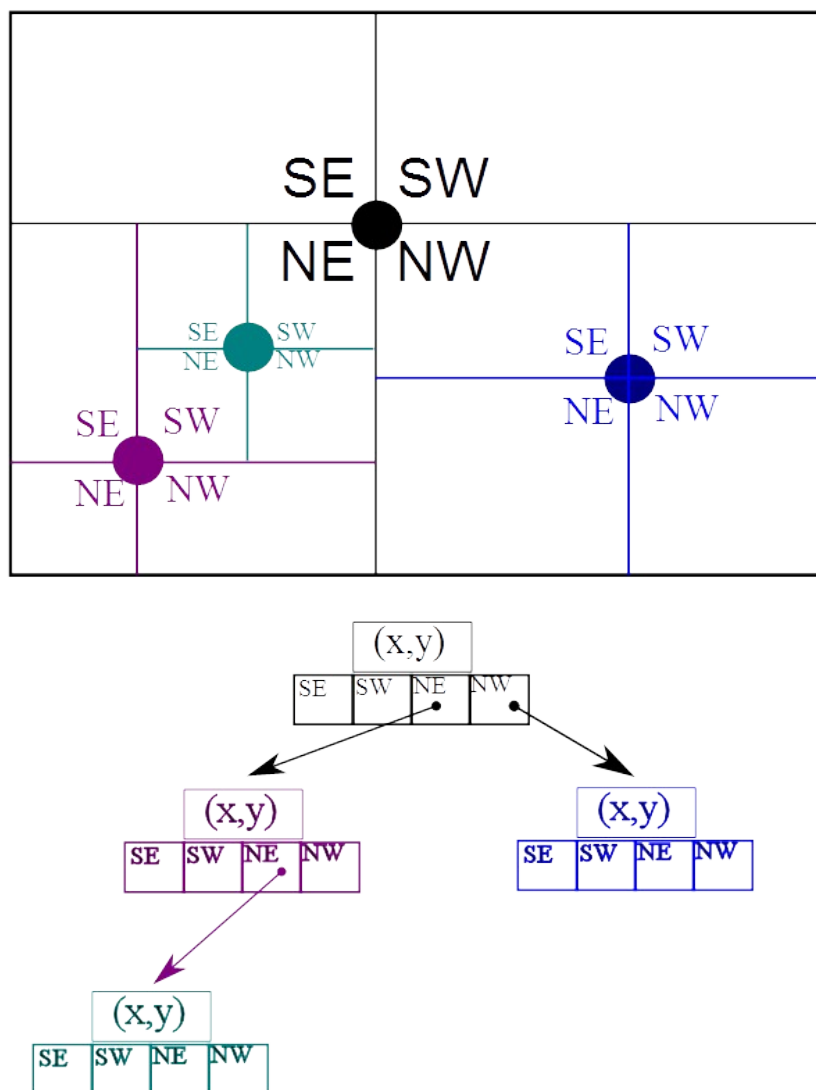
### *Point quadtree*

The point quadtree is an adaptation of a binary tree used to represent two-dimensional point data. It shares the features of all quadrees but is a true tree as the center of a subdivision is always on a point. It is often very efficient in comparing two-dimensional, ordered data points, usually operating in  $O(\log n)$  time.

Point quadrees are constructed as follows. Given the next point to insert, we find the cell in which it lies and add it to the tree. The new point is added such that the cell that contains it is divided into quadrants by the vertical and horizontal lines that run through the point. Consequently, cells are rectangular but not necessarily square. In these trees, each node contains one of the input points.

Since the division of the plane is decided by the order of point-insertion, *the tree's height is sensitive to and dependent on insertion order*. Inserting in a "bad" order can lead to a tree of height linear in the number of input points (at which point it becomes a linked-list). If the point-set is static, pre-processing can be done to create a tree of balanced height.

A figura abaixo mostra alguns pontos distribuídos no plano. Um ponto, ao ser inserido, divide uma região do plano em quatro regiões, a saber, sudeste, sudoeste, nordeste e noroeste. A figura também mostra a quadtree relativa àqueles pontos.



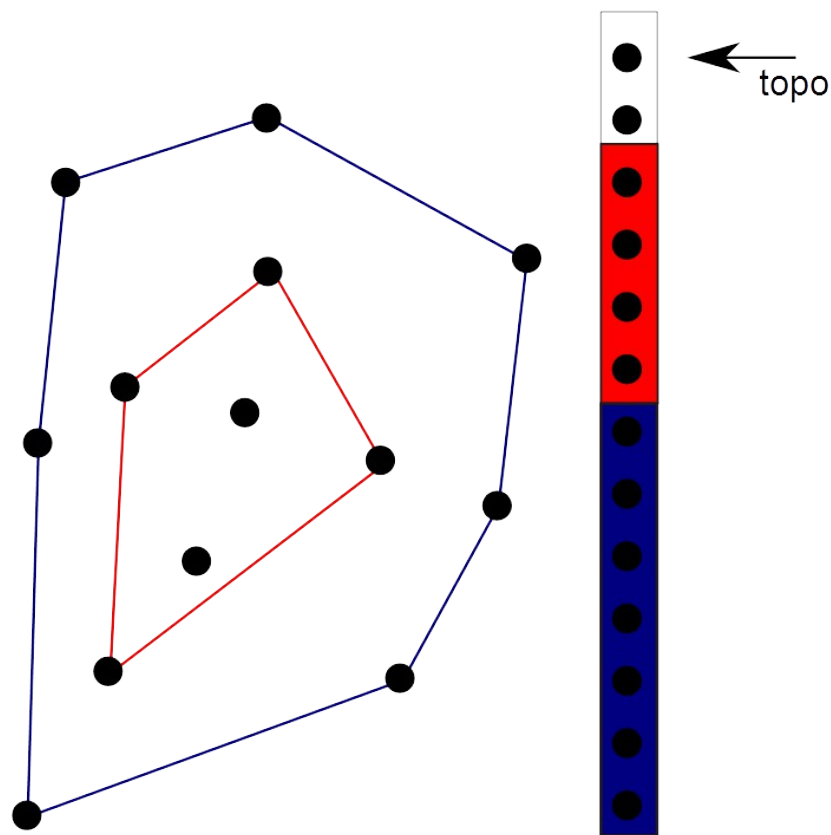
Como mencionado acima, a estrutura de uma quadtree depende da ordem de inserção dos pontos (no exemplo, o ponto preto foi o primeiro a ser inserido). Talvez a estratégia de inserção descrita a seguir possa produzir árvores razoavelmente balanceadas.

Considere a figura abaixo. O polígono azul é a envoltória convexa do conjunto de pontos. O polígono vermelho é a envoltória excluídos os pontos da primeira envoltória convexa, e assim sucessivamente. Note que, se inserirmos na quadtree os pontos na ordem inversa em que fizeram parte de alguma dessas envoltórias, provavelmente produzirá-se uma quadtree razoavelmente balanceada.<sup>14</sup>

O aluno deverá avaliar se esta estratégia de inserção é melhor (e em que medida) que simplesmente inserir as quadras e os equipamentos urbanos na ordem em que aparecem no arquivo

<sup>14</sup> Você concorda que esta estratégia tende a produzir árvores mais balanceadas? Como você imagina que seria a estrutura de uma quadtree de quadras se elas forem inseridas na ordem em que aparecem no arquivo .geo?

de entrada `.geo`. É necessário apresentar evidências para os argumentos apresentados (por exemplo, mostrar gráficos de testes com e sem estratégia de inserção sugerida)



## NOVOS COMANDOS E CONSULTAS

A seguir são mostradas duas tabelas com os novos comandos e novas consultas que são acrescentados aos arquivos `.geo` e `.qry`.

| comando                                     | parâmetros |   |
|---|------------|---|
| <b>hI</b>                                   | id vazao   | Informa a vazão (l/s) do hidrante <b>id</b> . O hidrante de código <b>id</b> deve ter sido inserido anteriormente na cidade por um comando <b>h</b> .     |
| <b>tI</b>                                   | id r       | Informa o raio (m) de alcance da radio-base <b>id</b> . A rádio-base <b>id</b> deve ter sido inserido anteriormente por um comando <b>t</b> .             |
| <b>sI</b>                                   | id t       | Informa o tempo (s) do ciclo (verde, amarelo, vermelho) do semáforo <b>id</b> . O semáforo deve ter sido inserido anteriormente por um comando <b>s</b> . |
| Novos comandos do arquivo <code>.geo</code> |            |   |



| comando                  | parâmetros              |   |
|--------------------------|-------------------------|---|
| <b>pc?</b>               | sufixo [ x y larg alt ] | <i>calcula o polígono de cobertura das rádio-bases da cidade (ou daquelas contidas na região retangular definida pelos parâmetros). Produz um arquivo svg mostrando a cidade e o polígono de cobertura. A área de cobertura de cada rádio-base deve estar evidenciada na saída svg. O nome do arquivo de saída deve ser uma composição do nome do arquivo .geo com o nome do arquivo .qry e do sufixo informado no comando:<br/><b>nomegeo-nomeqry-sufixo.svg</b></i> |
| <b>ac?</b>               | [ x y larg alt ]        | <i>calcula e imprime no arquivo .txt a área (m<sup>2</sup>) do polígono de cobertura das rádio-bases da cidade (ou daquelas contidas na região retangular definida pelos parâmetros).</i>   |
| Comandos do arquivo .qry |                         |   |

## A IMPLEMENTAÇÃO

Nesta fase, as quadras e os equipamentos urbanos devem ser armazenados em diferentes quadrees. Também, como comentado anteriormente, deve-se implementar um dos algoritmos de ordenação mencionados.

Como anteriormente, o módulos deve ser bem projetados e bem documentados. Escrever o .h antes de escrever o .c!!! Isto é muito importante!!

Lembrar que o arquivo `resumo.txt` deve continuar a ser produzido!

Como mencionado anteriormente, o aluno deverá justificar se a estratégia de inserção proposta é melhor do que inserir as quadras e os equipamentos urbanos na ordem em que aparecem no arquivo .geo. Para tal, provavelmente será necessário instrumentar o código para: ora executar a inserção com a estratégia de inserção proposta, ora executar os mesmos testes sem tal estratégia; contabilizar as operações sigficativas dos algoritmos de busca, inserção e delição dos dados geo-referenciados; entre outros. Uma vez coletados os dados, o aluno poderá fazer gráficos, etc, para tirar suas conclusões.

## FASE IV

A nossa cidade, até o momento, possui quadras, semáforos, hidrantes, torres de celular. Agora, a cidade receberá moradias, pessoas e estabelecimentos comerciais.

Além de possuir estabelecimentos comerciais comuns, a cidade possui duas operadoras de telefonia móvel, a SercomtUEL e a UELMobile. As operadoras precisam controlar as informações sobre seus clientes ativos, incluindo dados cadastrais e rádio-base a qual está correntemente conectado. Clientes das operadoras podem migrar para a concorrente. Devido a leis municipais, as operadoras não podem armazenar dados sobre algum cliente que tenha cessado de usar seus serviços.

Em vista deste cenário, o sistema deverá ser capaz de determinar, entre outros, quais aparelhos móveis estão conectados a quais rádio-bases; quem são os clientes correntes de cada uma das operadoras. Além disso, o sistema responderá a novas consultas, tais como:

- Quais são os moradores de uma quadra Q e seus telefones?
- Quais são os moradores de uma região R?
- Quais são as quadras que tem um estabelecimento do tipo T?

### ÍNDICES

A fim de facilitar algumas operações e consultas, o sistema utiliza índices. Para o nosso propósito, neste momento, índices são atalhos. Mais precisamente, são informações sobre os dados armazenados que facilitam sua localização. Por exemplo, a figura abaixo mostra duas tabelas. A tabela azul à esquerda (tabela principal) mostra uma agenda telefônica ordenada pelo nome. Neste caso, é fácil encontrar o telefone da "Guilhermina", devido à ordenação. O reverso, isto é, dado um telefone determinar o nome do assinante, é mais difícil.

A tabela à direita (amarela) é uma tabela auxiliar (índice). Ela contém os mesmos números de telefones da tabela principal e uma indicação da linha onde este telefone ocorre naquela tabela. Por exemplo, o telefone 32147530 (primeira linha da tabela amarela) ocorre na linha 5 da tabela principal; assim, determina-se facilmente que tal telefone é da Clarissa.

|    | NOME        | FONE     |    | FONE        | POS |
|----|-------------|----------|----|-------------|-----|
| 1  | Alvaro      | 13434365 |    | ** 32147530 | 5   |
| 2  | Ana         | 32315623 |    | 32315623    | 2   |
| 3  | Andre       | 56783456 |    | 32569872    | 20  |
| 4  | Carlos      | 43670987 |    | 43670987    | 4   |
| 5  | Clarissa    | 32147530 | ** | 56783456    | 3   |
| 6  | Domenica    | 98795432 |    | 58769987    | 7   |
| 7  | Eleutério   | 58769987 |    | 75640967    | 16  |
| 8  | Fabiana     | 78654320 |    | 76900098    | 17  |
| 9  | Fábio       | 76989876 |    | 76989876    | 9   |
| 10 | Guilherme   | 84563589 |    | 78654320    | 8   |
| 11 | Guilhermina | 80453587 |    | 80453587    | 11  |
| 12 | Hélio       | 91997204 |    | 83434365    | 1   |
| 13 | José        | 91676300 |    | 84563589    | 10  |
| 14 | Josefina    | 89099788 |    | 89099788    | 14  |
| 15 | Laura       | 89998742 |    | 89449922    | 19  |
| 16 | Marcos      | 75640967 |    | 89766677    | 18  |
| 17 | Maria       | 76900098 |    | 89998742    | 15  |
| 18 | Nair        | 89766677 |    | 91676300    | 13  |
| 19 | Norberto    | 89449922 |    | 91997204    | 12  |
| 20 | Ximena      | 32569872 |    | 98795432    | 6   |

## ARQUIVOS DE ENTRADA

A seguir, são apresentados novos arquivos de entrada que deverão ser processados

### Estabelecimentos Comerciais

| comando                       | parâmetros                  |  |
|-------------------------------|-----------------------------|--|
| t                             | codt descricao              | Define tipo de estabelecimento comercial   |
| e                             | codt cep face num cnpj nome | Insere um novo estabelecimento comercial de um determinado tipo (codt), localizado em um dado endereço (cep,face,num), que possui um dado cnpj e tem um dado nome. |
| Comandos do arquivo .ec (-ec) |                             |  |

### Pessoas e moradores

| comando | parâmetros |
|---------|------------|
|---------|------------|

|                                      |                              |   |
|--------------------------------------|------------------------------|---|
| <b>p</b>                             | cpf nome sobrenome sexo nasc | <i>Insere pessoa identificada por cpf, nomeada (nome,sobrenome), de um certo sexo (M,F), nascida numa determinada data (dd/mm/aaaa)</i> |
| <b>m</b>                             | cpf cep face num compl       | <i>Informa que um dada pessoa (cpf) mora num dado endereço (cep,face,num,compl)</i>   |
| <b>Comandos do arquivo .pm (-pm)</b> |                              |   |

### Clientes de Operadores de Telefonia Móvel

| comando                                    | parâmetros |  |
|--|------------|--|
| <b>su</b>                                  | cpf numcel | <i>Pessoa identificada por cpf é cliente da linha móvel numcel da SercomtUEL</i> |
| <b>um</b>                                  | cpf numcel | <i>Idem para UELMobile</i>   |
| <b>Novos comandos do arquivo .tm (-tm)</b> |            |  |

### NOVOS COMANDOS E CONSULTAS

| comando                               | parâmetros |   |
|---------------------------------------|------------|---|
| <b>su</b>                             | id         | <i>Informa que torre de celular de identificação id pertence à operadora SercomtUEL</i> |
| <b>um</b>                             | id         | <i>Informa que torre de celular de identificação id pertence à operadora UELMobile</i>  |
| <b>Novos comandos do arquivo .geo</b> |            |   |

| comando | parâmetros |  |
|---------|------------|--|
|---------|------------|--|

|            |              |  |
|------------|--------------|--|
| <b>mv</b>  | opdest num   | <p>Migra linha de celular (num) para operadora opdest (su ou um). Num deve existir na operadora de origem (cc, mensagem de erro). Caso o num já exista na operadora de destino, uma mensagem de erro deve ser apresentada e operação não é efetuada. Se o celular estiver conectado a uma rádio-base da antiga operadora, ele deve ser desconectado e reconectado na torre mais próxima da nova operadora (supor que não movimentou-se)</p> <p>SAIDA: arquivo .txt =&gt; texto “Celular <u>num</u>, pertencente a pessoa <u>nome</u>, migrou de <u>oporigem</u> para a <u>opdestino</u>. [Foi desconectado da rádio-base <u>id1</u> e reconectado na rádio-base <u>id2</u>]”</p> |
| <b>m?</b>  | cep          | <p>Moradores da quadra cujo cep é cep. Mostra mensagem de erro se quadra não existir.</p> <p>SAIDA: arquivo .txt =&gt; listar o nome, endereço, celular e operadora dos moradores da quadra.</p>   |
| <b>mr?</b> | x y larg alt | <p>Moradores das quadras que estão inteiramente dentro da região determinada.</p> <p>SAIDA: arquivo .txt =&gt; para cada quadra da região, produzir uma saída similar a da consulta m?.</p>  |
| <b>dm?</b> | cpf          | <p>Imprime todos os dados do morador identificado pelo cpf.</p> <p>SAIDA: arquivo .txt =&gt; dados pessoais, seu endereço, o número de seu celular (se houver) e respectiva operadora, e coordenada geográfica de onde mora.</p> <p>arquivo .svg =&gt; colocar um circulo (ou retângulo, ou outra forma geométrica) no lugar onde mora com seu cpf</p>   |
| <b>de?</b> | cnpj         | <p>Imprime todos os dados do estabelecimento comercial identificado por cnpj.</p> <p>SAIDA: arquivo .txt =&gt; dados do estabelecimento (nome, descrição do tipo, etc), endereço e coordenada geográfica.</p> <p>arquivo .svg =&gt; colocar um circulo (ou retângulo, ou outra forma geométrica) na coordenade geográfica do estabelimento.</p>  |

|             |                     |  |
|-------------|---------------------|--|
| <b>con</b>  | numcel cep face num | <p><i>Celular de número num conecta-se à rádio-base (da respectiva operadora) mais próxima. O usuário está no endereço cep/face/num.</i></p> <p><i>Obs. Não levar em consideração o raio de alcance da torre.</i></p> <p><i>SAIDA: arquivo .txt =&gt; Apresentar os dados do usuário (cpf, nome), a operadora do usuário, a identificação, a posicao e a distância da torre a qual se conectou.</i></p> <p><i>arquivo .svg =&gt; traçar uma linha entre a posição do usuário e a torre ao qual se conectou.</i></p>  |
| <b>mse?</b> | sx tp x y larg alt  | <p><i>Considere a região x,y,larg,alt. Quais são os moradores de uma quadra (dentro da região) que contenha um estabelecimento do tipo tp e que são do sexo sx (M ou F)?</i></p> <p><i>SAIDA: arquivo .txt =&gt; nome completo e endereço do morador.</i></p>  |
| <b>rip</b>  | cpf                 | <p><i>Infelizmente pessoa identificada por cpf morreu. Todos os dados armazenados sobre ela devem ser apagados (dados pessoais, dados sobre moradia e sobre sua linha móvel).</i></p> <p><i>SAIDA: arquivo .txt =&gt; imprimir os dados removidos de cada uma das estruturas. Por exemplo:</i></p> <p><i>RIP: José Silva, portador CPF, do sexo Masculino, nascido à 01/01/1920, residia no endereço CEP/NUM/FACE, usuário da linha móvel NUM da operadora SercomtUEL.</i></p> <p><i>arquivo .svg: colocar um losango preto, com um cruz branca no interior na residência do defunto.c</i></p> |
| <b>lk?</b>  | id                  | <p><i>Quais são os celulares conectados à rádio-base identificada por id?</i></p> <p><i>SAIDA: arquivo .txt=&gt; listar (enumerar) o número dos celulares correntemente ligados à rádio-base.</i></p> <p><i>arquivo .svg =&gt; mostrar no mapa a rádio-base em questão e mostrar a posição dos celulares conectados a ela.</i></p>   |
| <b>rb?</b>  | numcel              | <p><i>Em que rádio-base o celular numcel está conectado?</i></p>   |
| <b>co?</b>  | (su   um) [ n ]     | <p><i>Mostra, em ordem alfabética, o nome e o número do celular dos clientes de uma dada operadora de telefonia móvel. Lista os n primeiros nomes.</i></p> <p><i>SAIDA: arquivo .txt =&gt; mostra os dados de um cliente por linha</i></p>   |

|              |                          |   |
|--------------|--------------------------|---|
| <b>lnr?</b>  | x y larg alt [ (su um) ] | Mostra os celulares conectados as rádio-bases dentro da região delimitada pelo parâmetros x, y, larg, alt, da, se presente, operadora Sercomtuel (su) ou UELMobile (um).<br>SAIDA: arquivo .txt => listar, para cada radio-base, os celulares conectados. |
| <b>ecq?</b>  | cep                      | Lista os estabelecimentos de uma determinada quadra.<br>SAIDA: arquivo .txt => mostrar cnpj, nome, tipo e endereço de cada estabelecimento comercial.   |
| <b>ecr?</b>  | tp [ x y larg alt ]      | Lista os estabelecimentos comerciais de um da tipo de, se presente, uma determinada região.   |
| <b>tecq?</b> | cep                      | Lista os tipos de estabelecimentos comerciais de uma dada quadra.<br>SAIDA: arquivo .txt => mostrar, para cada tipo de estabelecimento comercial, listar o nome dos estabelecimentos daquele tipo.  |
| <b>tecr?</b> | x y larg alt             | Quais são os tipos de estabelecimentos comerciais existentes numa dada região.  |
| <b>dc?</b>   | num                      | Quem é o usuário da linha móvel de número num?<br>SAIDA: arquivo .txt => mostrar dados pessoais do dono e nome da operadora .   |
| <b>lec?</b>  | num                      | Onde está (endereço) o celular de número num?<br>SAIDA: arquivo .txt => mostrar dados pessoais do dono do celular e o endereço de onde o celular foi conectado pela última vez.<br>arquivo .svg => mostrar no mapa a localização do celular               |
| <b>lcc?</b>  | num                      | Onde está (coordenada) o celular de número num? Semelhante à consulta lec?, porém, reporta a coordenada geográfica da localização do celular.   |
| <b>dpr</b>   | x y larg alt             | Desapropria região.<br>Apagar quadras, hidrantes, etc dentro da região; desconectar celulares; etc<br>SAIDA: arquivo .txt => listar o que foi removido<br>arquivo .svg => elementos removidos não devem aparecer.   |

### Novos comandos do arquivo .qry

#### SAIDA

ESCREVER

#### IMPLEMENTAÇÃO

Deve ser implementado um módulo para o tipo abstrato Tabela de Espalhamento. A

implementação da tabela de espalhamento deve fazer uso da lista implementada (conforme solicitado) nos trabalhos anteriores.

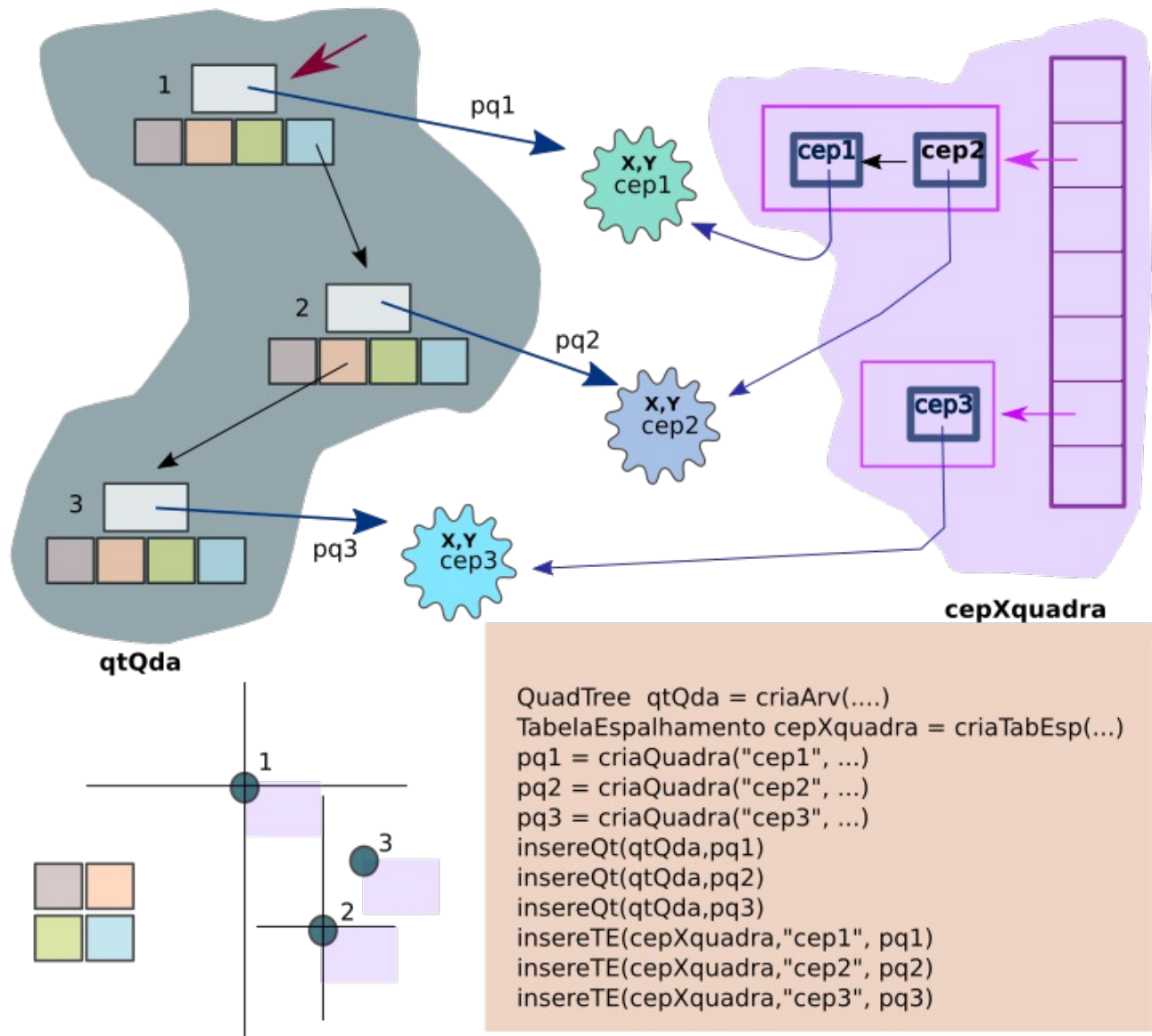
As quadras e os equipamentos urbanos devem continuar a ser armazenados em quadtrees. Tabelas de espalhamento devem ser usadas como dicionários e como índices. Um dicionário, por exemplo, pode (deve) ser usado para manter a tabela de tipos de estabelecimentos comerciais (tipo X descrição). Vários índices deverão ser usados para facilitar as consultas.

Índices/Dicionários que devem ser mantidos e usados em consultas:

- cpf X cep: dado o cpf, retorna o dep quadra onde a pessoa reside
- numcel X pessoa: dado o número de um celular retorna o cpf do dono do celular
- numcel X rádio-base: dado o número de um celular, retorna rádio-base a qual celular está conectado
- tipo estabelecimento comercial X descrição: dado o código do tipo do estabelecimento, retorna a descrição
- dados de uma pessoa: dado um cpf, retorna os dados pessoais daquela pessoa
- cep X quadra: dado um cep, retorna os dados da respectiva quadra

**ATENÇÃO:** Espera-se que partes comuns, principalmente , para a solução das consultas estejam fatorados (i.e., divididos em procedimentos curtos que executem uma única tarefa) e que sejam reutilizados largamente.







## FASE V (Rascunho)

A nossa cidade está completa. Possui quadras, equipamentos urbanos, moradores, e estabelecimentos comerciais. Agora, queremos determinar trajetos na cidade. Uma possível consulta seria:

*"Qual é o melhor (mais curto, mais rápido) entre o endereço relativo à última posição conhecida do telefone F e o endereço X?"*.

O resultado das consultas podem ser textuais ou podem ser pictóricas. Um exemplo de uma resposta textual poderia ser:

*Siga na direção norte na Rua Xxxx até o cruzamento com a Rua Yyyy. Siga na Rua Yyyy na direção sul.....*

O resultado pictórico é mostrado no arquivo .svg produzido evidenciando o caminho a ser percorrido (exemplo, Figura 2).

### Entrada de Dados

A entrada de dados será feita por meio dos arquivos-texto do trabalho anterior, de um novo arquivo texto (**arquivo.via**) com a descrição do sistema viário da cidade . A maior parte das novas consultas indagam sobre o melhor percurso (menor distância ou menor tempo estimado) entre uma origem e um destino.

A origem e o destino são referências geográficas. Referências geográficas são obtidas por meio dos comandos iniciados por @ e armazenadas nos registradores R0 .. R10. A referência geográfica armazenada num registrador pode ser utilizada em outras consultas.

A resposta esperada pode ser a descrição textual do percurso ou a representação pictórica do percurso. No caso da representação pictórica, na consulta é informado o sufixo do arquivo de saída (**nome-arq-sufixo.svg**).


A sintaxe destas consultas são da forma:

- comando (t|p sufixo) (D|T) **outros-parâmetros**
  - D: menor distância
  - T: menor tempo estimado
  - t: descrição textual
  - p: representação pictórica
  - sufixo: sufixo do arquivo de saída

Existem ainda outras consultas cujas respostas não são percursos e, portanto, não necessitam dos parâmetros básicos mostrados acima.

Os novos comandos e seus parâmetros específicos estão listados na tabela abaixo. Os parâmetros básicos de percurso estão indicados por (¶).

| comando | parâmetros |  |
|---------|------------|--|
|---------|------------|--|

|                                |   |   |
|--------------------------------|---|---|
| @f?                            | r fone  | Armazena no registrador <i>r</i> a posição geográfica endereço onde o telefone <i>fone</i> foi conectado por último   |
| @m?                            | r cpf   | Armazena no registrador <i>r</i> a posição geográfica da residência do morador de <i>cpf</i>  |
| @e?                            | r cep face num  | Armazena no registrador <i>r</i> a posição geográfica do endereço <i>cep/face/num</i>   |
| @g?                            | r id  | Armazena no registrador <i>r</i> a posição geográfica do equipamento urbano cujo identificador é <i>id</i>  |
| @xy                            | r x y   | Armazena no registrador <i>r</i> a posição geográfica ( <i>x,y</i> )  |
| @tp?                           | r1 tp r2  | Armazena no registrador <i>r1</i> o estabelecimento comercial mais próximo da posição armazenada no registrador <i>r2</i>   |
| p?                             |  r1 r2 [cor] | Qual o melhor trajeto entre a origem está que no registrador <i>r1</i> e o destino que está no registrador <i>r2</i> . Caso o percurso peça a representação pictórica, o trajeto deve ser pintado com a cor especificada. |
| Novos comandos do arquivo .qry |   |   |

## O Mapa Viário

O mapa viário é um grafo direcionado: os vértices representam os extremos de um segmento de rua e os arcos representam um segmento de rua e indicam o sentido do tráfego.

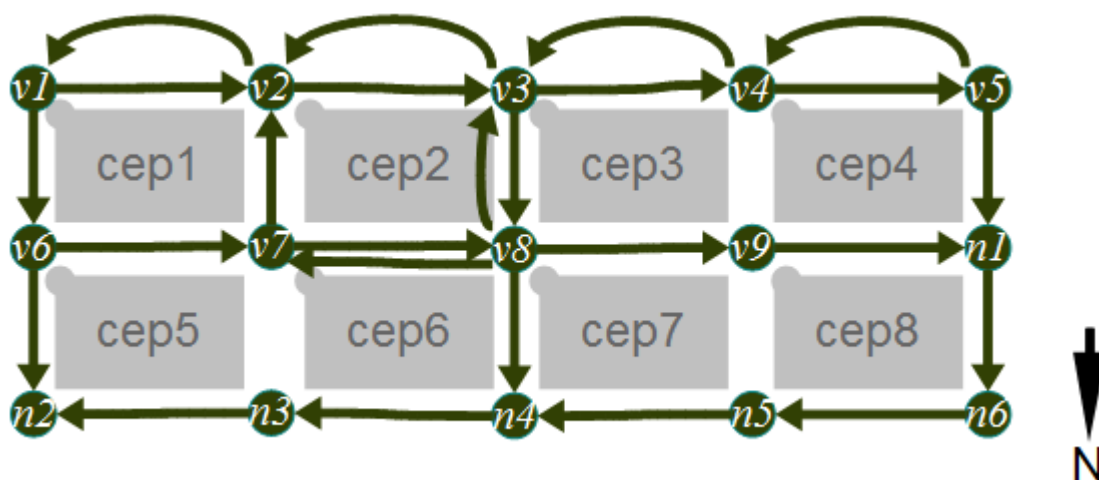


Figura 2:

Os vértices e os arcos possuem alguns atributos:

| Atributo        | Descrição   |
|-----------------|---|
| <b>Vertices</b> |   |
| id              | Um identificador  |
| x,y             | Uma coordenada do segmento de rua                                   |
| <b>Arestas</b>  |   |
| nome            | Nome da rua a qual pertence o segmento                              |
| ldir            | Cep da quadra que está do lado direito do segmento de rua           |
| lesq            | Idem para o lado esquerdo   |
| cmp             | comprimento (em metros) do segmento de rua                          |
| vm              | Velocidade média (m/s) que os carros trafegam neste segmento de rua |

### O Arquivo do Mapa Viário

O arquivo do mapa viário possui o seguinte formato (os campos são separados por um espaço em branco):

|                             |   |
|-----------------------------|---|
| v id x y                    | <i>cria o vértice id posicionado nas coordenadas [x,y]</i>  |
| e i j ldir lesq cmp vm nome | <i>cria a aresta (i,j) e associa as outras informações à aresta. Caso a aresta não possua quadras em algum de seus lados, esta ausência é indicada por um hífen (-)</i> |

Abaixo, um exemplo deste arquivo:

```
v v1 10 10
v v2 110 10
v v3 210 10
v v4 310 10
v v5 410 10
v v6 10 70
v v7 110 70
v v8 210 70
...
e v1 v6 - cep1 70.0 3.5 Rua_Belo_Horizonte
e v7 v8 cep6 cep2 100.0 4.0 Av_Higienopolis
e v8 v7 cep2 cep6 100.0 5.0 Av_10_de_Dezembro
...
mapa-viario1.via
```

### A Implementação

Para a determinação de caminhos mínimos deve ser utilizado o algoritmo de Dijkstra. O TAD de grafo direcionado apresentado em aula deve ser completamente implementado com algumas modificações que se façam necessárias.

Vale relembrar e enfatizar as as diversas estruturas de dados (árvore, grafo, tabela de espalhamento, etc) devem ser implementadas em módulos separados e bem documentados.

Para facilitar a implementação, alguns fatos podem ser considerados:

- Existem vértices do grafo posicionado no ponto médio das intersecções de ruas.
- Normalmente, existirão vértices no ponto médio de um segmento de rua. Isto é, o segmento de rua referente a uma face de uma quadra poderá ser, na prática, representado por duas aresta: (esquina1,meioquadra) e (meioquadra,esquina2). A figura abaixo ilustra onde serão posicionados os vértices extras (em lilás) Isto não altera o cálculo do percurso, apenas facilita posicionar o início e o fim do percurso.

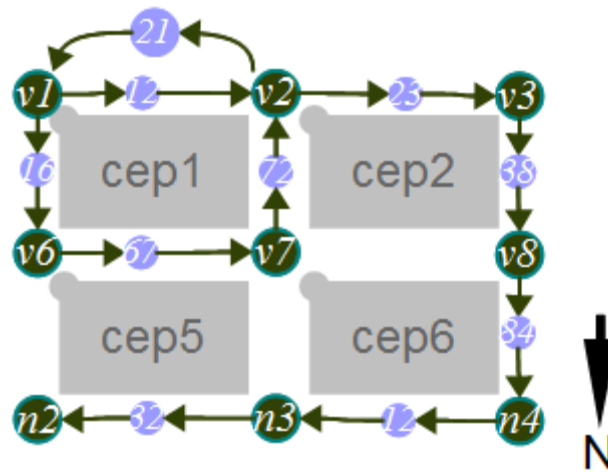


Figura:

## A Saída

A execução e a saída deverão produzir as mesmas do trabalho anterior. Além disso, (a) a representação pictórica de um percurso deve apenas apresentar as quadras vizinhas ao percurso.<sup>15</sup> Por exemplo, na figura abaixo o resultado da consulta será apenas a parte nítida do mapa; (b) a resposta textual da consulta deve ser colocada no arquivo .txt e deve ser precedida (como um título) pelo texto (linha) da consulta; (c) ao final do processamento do arquivo .geo, o .svg gerado deve incluir o mapa viário (grafo) sobreposto ao mapa da cidade.

<sup>15</sup> Dica: determine as coordenadas mínimas e máximas referentes ao percurso e use-as como região de busca para determinar as quadras vizinhas



## RESUMO DOS PARÂMETROS DO PROGRAMA SIGUEL

| Parâmetro / argumento | Opcional | Descrição   |
|-----------------------|----------|---|
| -acc0                 | S        | Início da bateria de testes   |
| -acc                  | S        | Continuação da bateria de testes  |
| -e <i>path</i>        | S        | Diretório-base de entrada ( <b>BED</b> )  |
| -f <i>arq.geo</i>     | N        | Arquivo com a descrição da cidade. Este arquivo deve estar sob o diretório <b>BED</b> . |
| -id                   | S        | Se presente, deve imprimir na saída padrão o nome e a matrícula do aluno                |
| -n <i>num</i>         | X        | <b>Descontinuado</b>  |
| -o <i>path</i>        | N        | Diretório-base de saída ( <b>BSD</b> )  |
| -q <i>arq.qry</i>     | S        | Arquivo com consultas   |
| -ec <i>arq.ec</i>     | S        | Arquivo dos estabelecimentos comerciais   |
| -pm <i>arq.pm</i>     | S        | Arquivo das pessoas e moradores   |
| -tm <i>arq.tm</i>     | S        | Arquivo dos clientes de telefonia móvel   |
| -v <i>arq.via</i>     | S        | Arquivo de vias   |

**ATENÇÃO:** novos parâmetros acrescentados