MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

CAP-349 – Bancos de Dados Geográficos

Aplicação de Rede em Banco de Dados Geográfico PostGIS

Fernando Bagnara Mussio
fmussio@gmail.com

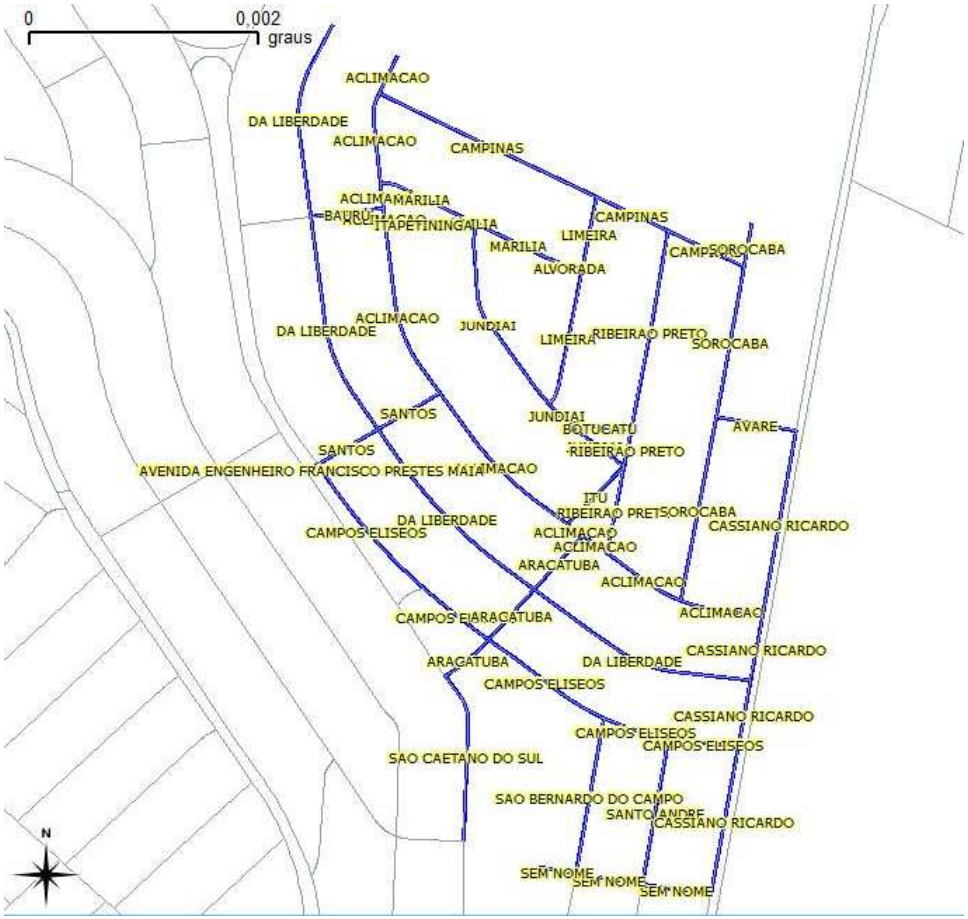
3 de Setembro de 2010

Objetivos

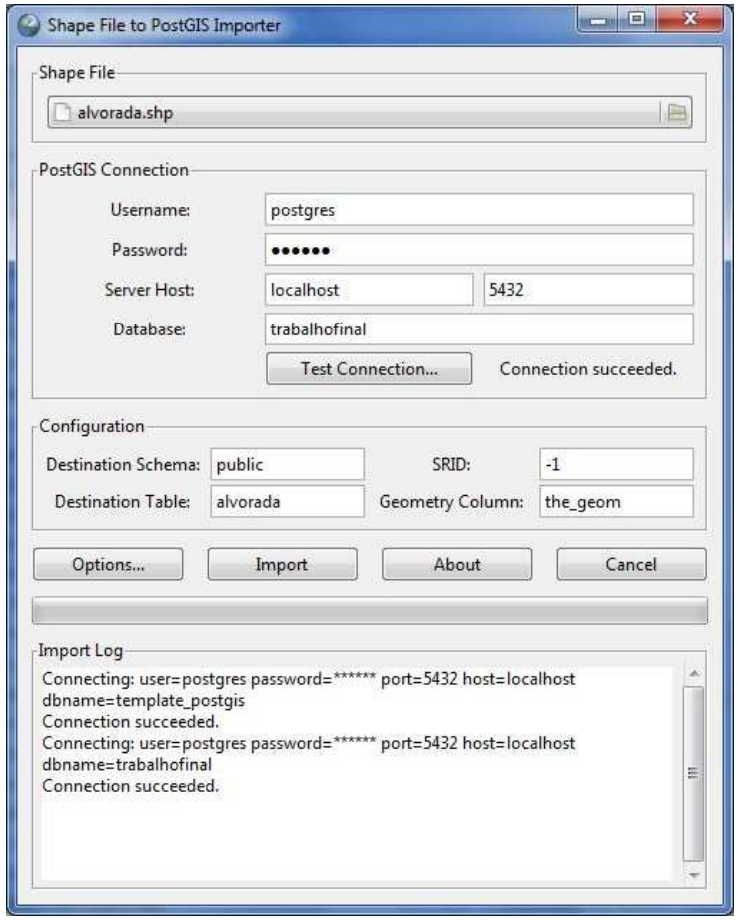
- Criar um modelo de rede para armazenar dados de arruamento;
- Utilizar o banco de dados PostgreSQL com extensão PostGIS para armazenar os dados, fazer a importação de arquivos ESRI Shapefile e processar geometrias;
- Criar uma aplicação exemplo capaz de recuperar e visualizar os dados do modelo de dados criado;

Importação do arquivo ESRI Shapefile

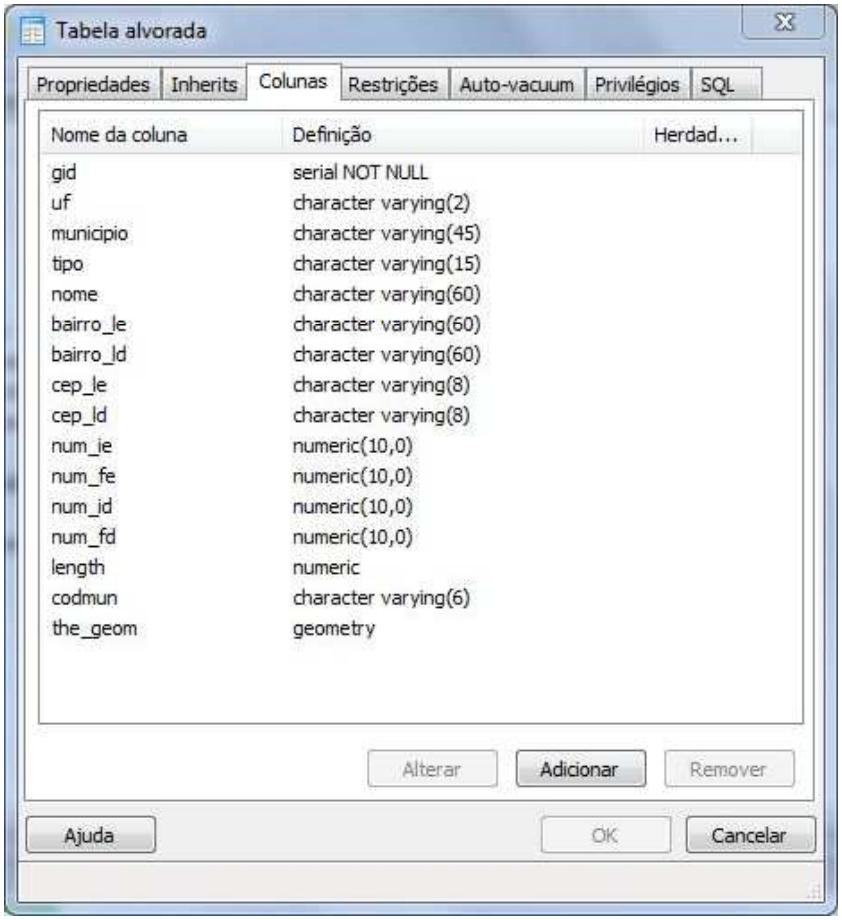
Seleção do mapa vetorizado



Importação do arquivo ESRI Shapefile



Importação do arquivo ESRI Shapefile



Eliminar GEOMETRYCOLLECTION

- As ruas do ShapeFile tem geometria descrita em MULTILINESTRING

  SELECT ST_AsText(the_geom) FROM alvorada;

  Exemplo de resultado: "MULTILINESTRING((-45.9146741183431 -23.2197785460771,-45.9148251721218 -23.2200902639533))“

- Muitas operações topológicas do PostGIS não suportam GEOMETRYCOLLECTION.
- A função ST_Dump quebra uma GEOMETRYCOLLECTION em GEOMETRY

  SELECT ST_AsText((ST_Dump(the_geom)).geom) FROM alvorada;

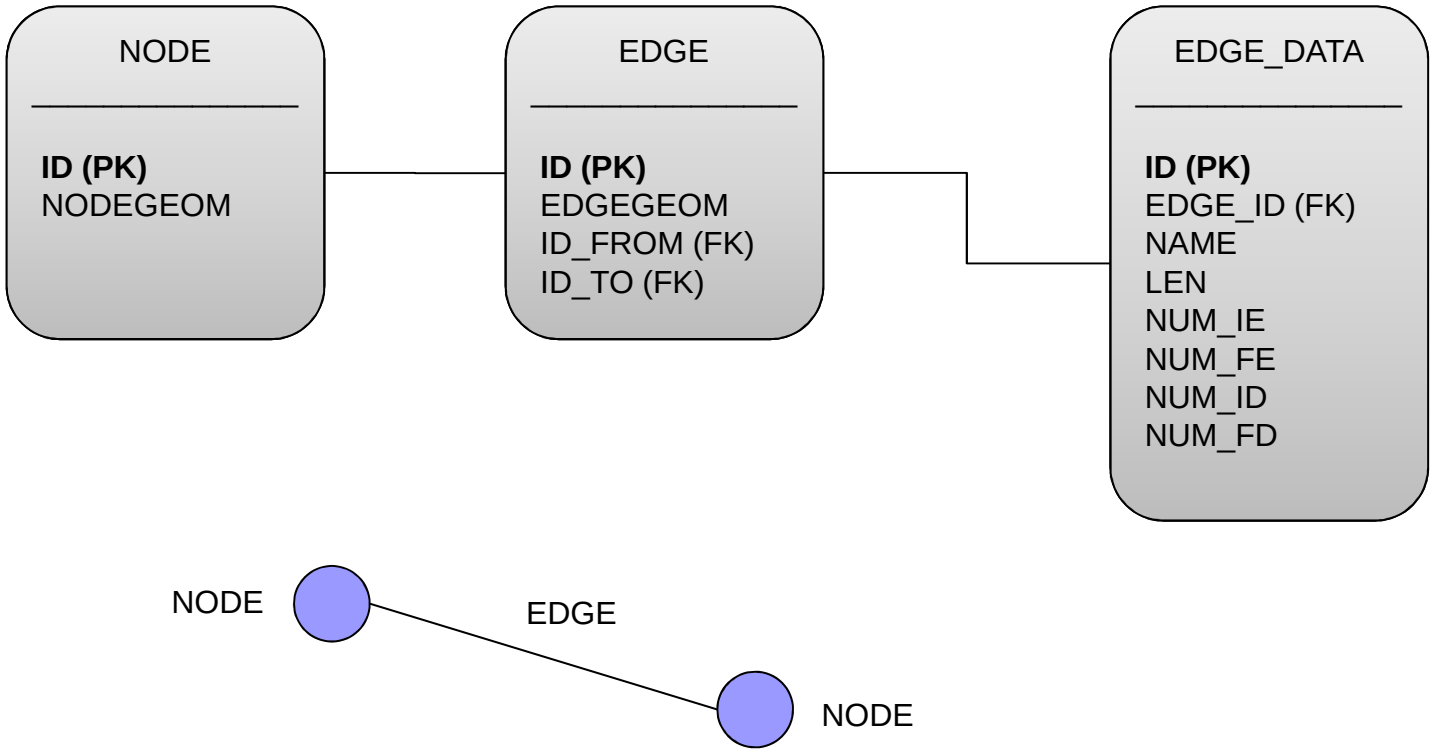
  Exemplo de resultado: "LINESTRING(-45.9146741183431 -23.2197785460771,-45.9148251721218 -23.2200902639533)"

Mais Pré-Processamento

- Por quê isto foi necessário?

```
UPDATE
        alvorada
SET
        dumped_geom = ST_GeometryFromText(
                ST_AsText(dumped_geom)
        );
```

Diagrama do Modelo de Dados



Criar Tabela de NÓS

- Contém a localização dos cruzamentos de ruas.

```
CREATE TABLE NODE (
        ID SERIAL PRIMARY KEY,
        NODEGEOM GEOMETRY
);
```

Criar Tabela de ARESTAS

- Contém a geometria e estrutura das ruas.

```
CREATE TABLE EDGE (
        ID SERIAL PRIMARY KEY,
        EDGEGEOM GEOMETRY,
        FROM_ID INT,
        TO_ID INT
);
```

Criar Tabela de Dados da ARESTA

- Contém dados adicionais das ruas.

```
CREATE TABLE EDGE_DATA (
        ID SERIAL PRIMARY KEY,
        EDGE_ID INT,
        NAME VARCHAR(60),
        NUM_IE INT,
        NUM_FE INT,
        NUM_ID INT,
        NUM_FD INT,
        LEN NUMERIC
);
```

Processamento dos NÓS

- Usar as funções ST_StartPoint e ST_EndPoint para obter uma lista dos pontos que sejam início/fim de ruas e avenidas, estes pontos serão NÓS do grafo.

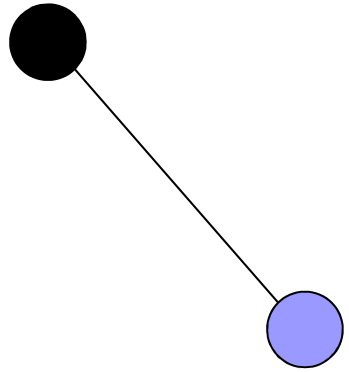
```
SELECT DISTINCT ST_StartPoint(the_geom)
AS startpoint   FROM alvorada;

SELECT ST_EndPoint(the_geom) AS endpoint FROM alvorada
        WHERE ST_EndPoint(the_geom) NOT IN (
                SELECT DISTINCT ST_StartPoint(the_geom)
                AS startpoint FROM alvorada
        );
```

Processamento das ARESTAS

- Para cada NÓ de início distinto encontrado (**startpoint**), buscar as arestas que se conectam àquele determinado NÓ e o NÓ da outra extremidade;

```
SELECT
        dumped_geom AS aresta,
        ST_EndPoint(dumped_geom) AS endpoint,
        name, num_ie, num_fe, num_id, num fd
FROM
        alvorada
WHERE
        ST_Equals([startpoint]::Geometry,
        ST_StartPoint(dumped_geom) );
```



[startpoint] é o WKT de um ponto, poderia ser, por exemplo 'POINT(-45.9118829606849 -23.2229230218845)'

Processamento das ARESTAS

- Buscar o ID do nó endpoint na tabela NODE, se não existir, criar o nó e guardar o valor do ID para referenciá-lo no campo NODE_TO da tabela EDGE.

```
SELECT ID AS id_endpoint FROM NODE WHERE
    ST_Equals(NODEGEOM, [endpoint]::Geometry);

INSERT INTO NODE (NODEGEOM)
    VALUES ( [endpoint]::Geometry);
```

(depois repetir o SELECT para buscar o ID)

Processamento das ARESTAS

- Inserir os dados da aresta encontrada dos campos da tabela EDGE.

```
INSERT INTO EDGE
      (EDGEGEOM, FROM_ID, TO_ID)
VALUES
      ( [aresta]::Geometry, [startpoint_id], [endpoint_id]);
```

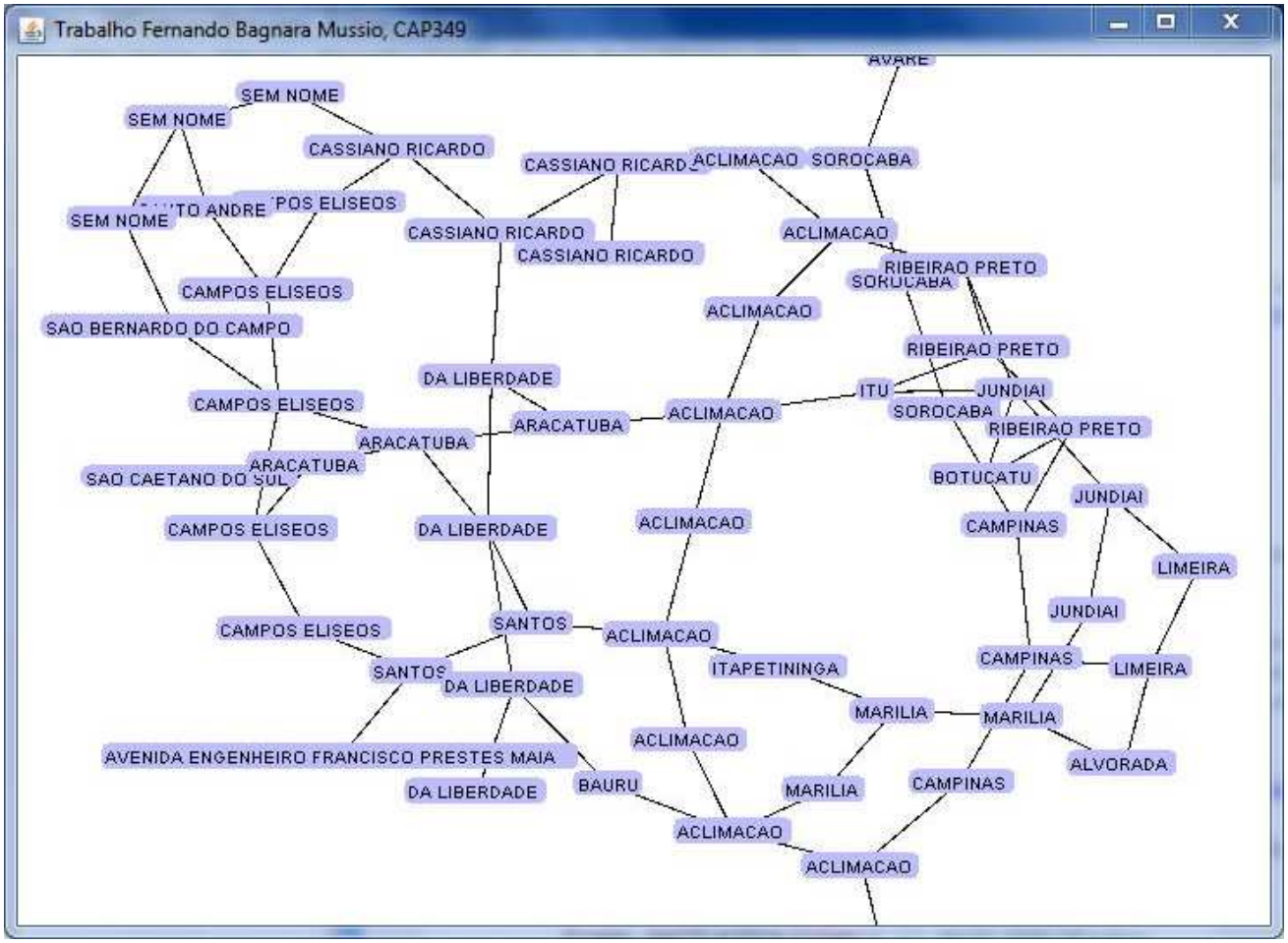
(depois fazer um SELECT para buscar o ID)

Processamento das ARESTAS

- Inserir os dados da aresta encontrada dos campos da tabela EDGE_DATA.

```
INSERT INTO EDGE_DATA
        (EDGE_ID, NAME, NUM_IE, NUM_FE, NUM_ID, NUM_FD, LEN)
VALUES
        ([edge_id], [num_ie], [num_fe], [num_id], [num_fd], [len]);
```

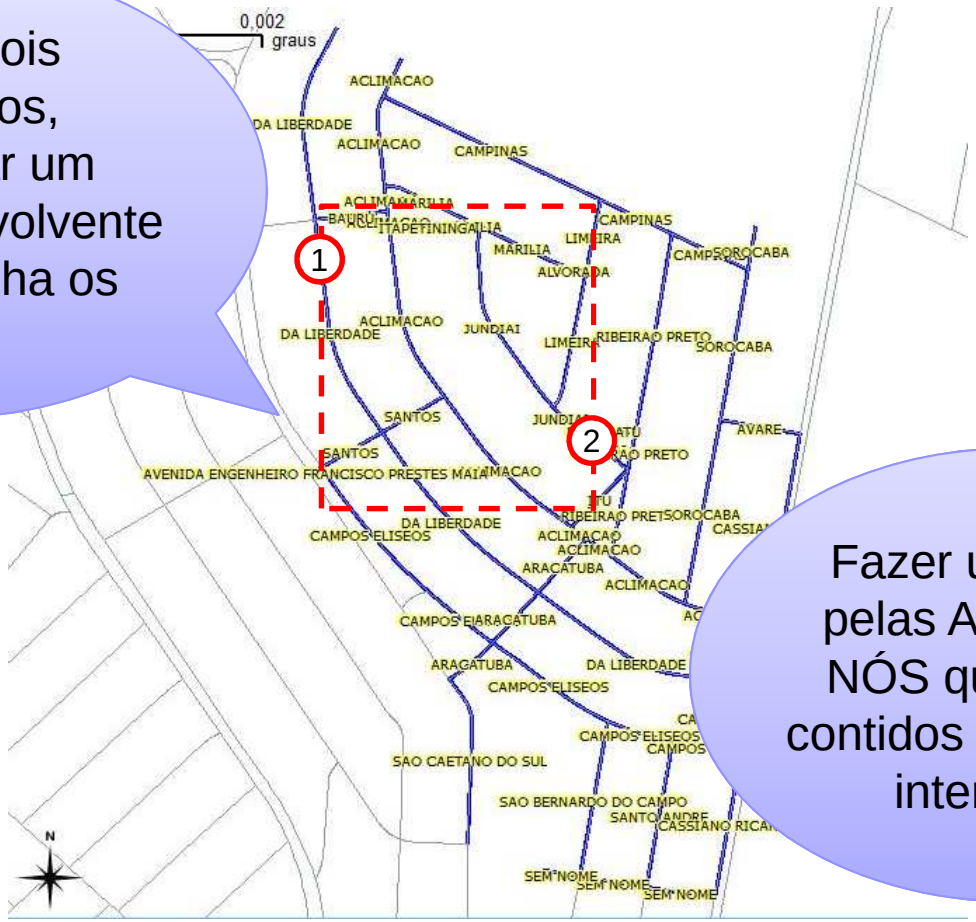
Recuperação e Visualização da Rede



http://www.prefuse.org/

Exemplo de Aplicação

Dados dois endereços, determinar um retângulo envolvente que contenha os dois.



Fazer uma busca pelas ARESTAS e NÓS que estejam contidos na região de interesse…

Exemplo de Aplicação

- Endereço: CAMPOS ELISEOS, 717

```
SELECT ST_AsText(EDGEGEOM) AS address
FROM EDGE JOIN
EDGE_DATA ON EDGE.ID = EDGE_DATA.EDGE_ID
WHERE
EDGE_DATA.NAME = 'CAMPOS ELISEOS' AND (
    ((NUM_IE < 717) AND (NUM_FE > 717)) OR
    ((NUM_ID < 717) AND (NUM_FD > 717))
);
```

Exemplo de Aplicação

- Tendo buscado a geometria dos dois endereços, fazer a busca pelos elementos (NÓS e ARESTAS) contidos no retângulo envolvente das duas geometrias.

```
SELECT * FROM NODE WHERE ST_Contains(
ST_Envelope(ST_Union(
ST_GeomFromText('LINESTRING(-45.91……21094)'),
ST_GeomFromText('LINESTRING(-45.912……3715)')
)), NODEGEOM);
```

(fazer o mesmo para EDGE)

Exemplo de Aplicação

Edsger Wybe Dijkstra

Nascimento	11 de Maio de 1930 Roterdã, Países Baixos
Morte	6 de Agosto de 2002 (72 anos) Nuenen, Países Baixos
Nacionalidade	Neerlandês
Campo(s)	Ciência da computação
Conhecido(a) por	Algoritmo de Dijkstra Semáforo THE
Prêmio(s)	Prêmio Turing (1972)

- Reconstruir o grafo parcial a partir dos elementos encontrados.
- Buscar pelo menor caminho entre dois endereços no grafo, utilizando por exemplo o algorítmo de Dijkstra.

Referências

- High Performance Multimodal Networks
  Erik G. Hoel, Wee-Liang Heng, Dale Honeycutt

- REDES EM SISTEMAS DE INFORMAÇÃO GEOGRÁFICA, Gilberto Ribeiro de Queiroz

- PostGIS Manual1.5.1