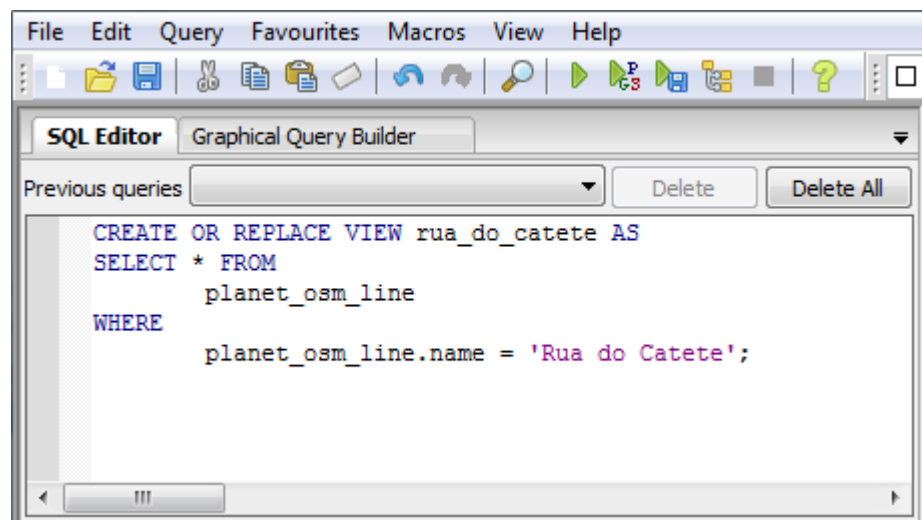


Trabalhando com rotas nos dados do OpenStreetMap: Parte 2

No [artigo anterior](#), criamos a tabela de topologia e aprendemos alguma coisa sobre ela e um pouco sobre como as rotas são armazenadas. Neste artigo vou me aprofundar um pouco mais nestes conceitos antes de prosseguir com a prática.

Vou usar o GeoServer para criar a visualização de certos dados da tabela *osm_2po_4pgr*. Infelizmente não é o escopo deste artigo ensinar a usar o GeoServer. Por sorte, tudo que vou exibir neste artigo é ilustrativo e servirá apenas para você acompanhar minhas explicações. Não há necessidade de executar nada por enquanto.

Vou começar criando uma *view* no nosso banco de dados *osm* somente para exibir a Rua do Catete (no Rio de Janeiro). Esta *view* não tem finalidade prática no cálculo de rotas e só servirá como ilustração dos conceitos mostrados no artigo:



Agora, é necessário criar uma SQL View no GeoServer. Vá em “Camadas”, selecione “Adicionar Novo Recurso”, escolha o *workspace* do nosso banco de dados OSM e clique em “Configure New SQL View”.

nd types. [Create new feature type...](#)
statement. [Configure new SQL view...](#)
a capa que deseja configurar



Acción
Publicar novamente
Publicar novamente
Publicar novamente

Na instrução SQL, preencha com

```
select * from public.rua_do_catete
```

clique em “Atualizar” e modifique os valores do atributo “way” conforme figura abaixo (tipo “LineString” e SRID “900913”:

way_area	Float
abandoned:aeroway	String
abandoned:amenity	String
abandoned:building	String
abandoned:landuse	String
abandoned:power	String
area:highway	String

way

LineString

900913

Guardar

Cancelar

Atualize o “*Bounding Box*” (Retângulos Envolventes) e salve. Eu criei um estilo para esta camada somente para que eu possa acompanhar os identificadores dos segmentos da rua.

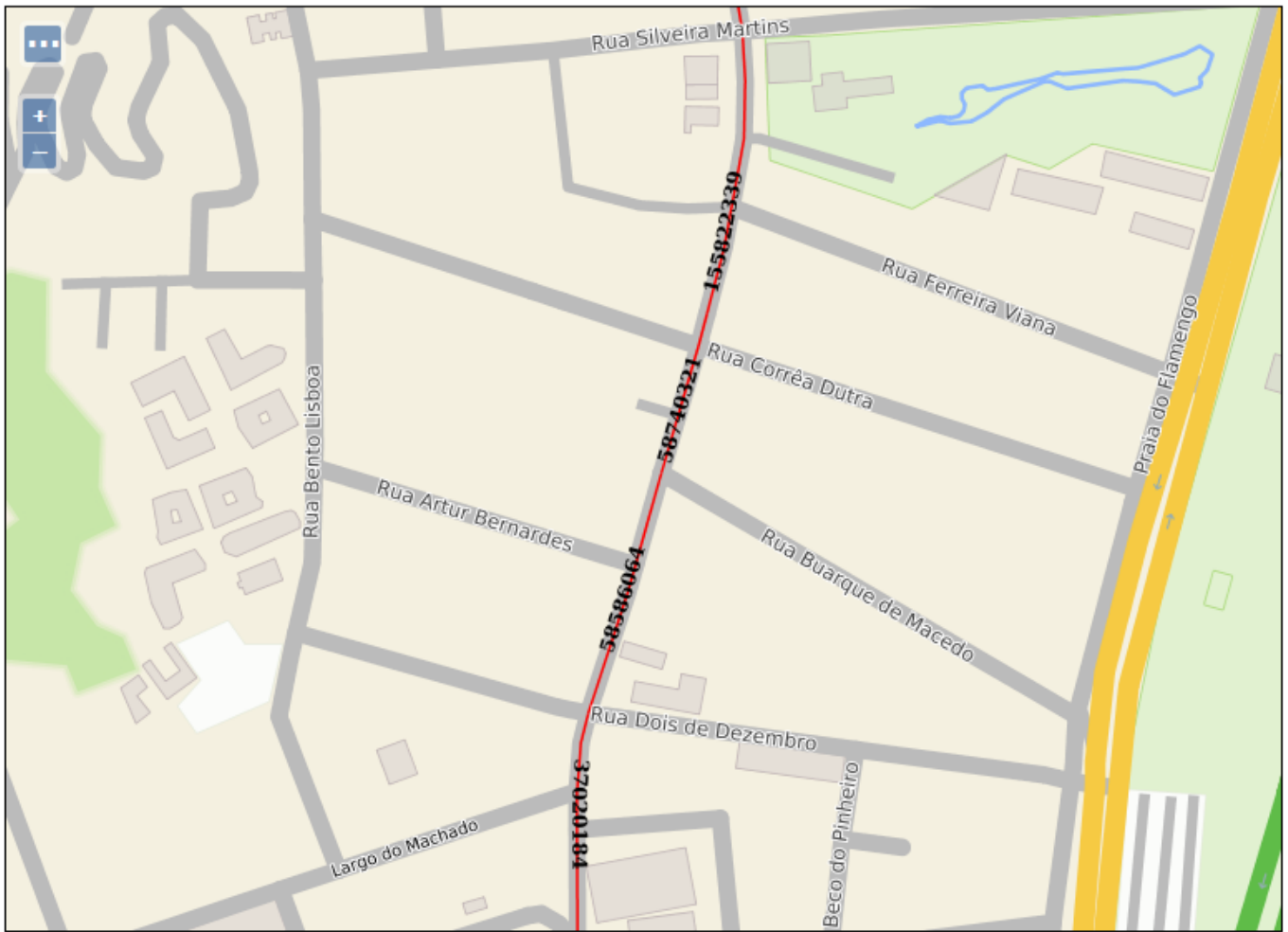
```
<NamedLayer>  
  <Name>catete_line</Name>  
  <UserStyle>  
    <Title>Estilo para a Rua do Catete</Title>
```

```

<FeatureTypeStyle>
  <Rule>
    <LineSymbolizer>
      <Stroke>
        <CssParameter name="stroke">#FF0000</CssParameter>
      </Stroke>
    </LineSymbolizer>
    <TextSymbolizer>
      <Label>
        <ogc:PropertyName>osm_id</ogc:PropertyName>
      </Label>
      <LabelPlacement>
        <LinePlacement />
      </LabelPlacement>
      <Fill>
        <CssParameter name="fill">#000000</CssParameter>
      </Fill>
      <Font>
        <CssParameter name="font-family">Arial</CssParameter>
        <CssParameter name="font-size">12</CssParameter>
        <CssParameter name="font-style">normal</CssParameter>
        <CssParameter name="font-weight">bold</CssParameter>
      </Font>
      <VendorOption name="followLine">true</VendorOption>
      <VendorOption name="maxAngleDelta">90</VendorOption>
      <VendorOption name="maxDisplacement">400</VendorOption>
      <VendorOption name="repeat">150</VendorOption>
    </TextSymbolizer>
  </Rule>
</FeatureTypeStyle>
</UserStyle>
</NamedLayer>

```

O próximo passo foi criar um grupo de camadas com a camada da Rua do Catete e o grupo de camadas do OSM (grupo que eu criei com todas as camadas do OSM). O resultado disso tudo você vê na figura abaixo:



É claro que você não precisaria de nada disso. Bastava selecionar os registros na tabela com o SQL usado pela *view* e pronto, mas a princípio não daria para saber a sequência correta dos segmentos e quem conecta com quem, pois esta informação está na geometria (e na tabela de topologia, mas para achar alguma coisa lá precisamos saber o que estamos procurando). Já temos os segmentos da Rua do Catete com seus respectivos identificadores. Agora podemos procurar um segmento da rua na tabela de topologia. É melhor considerar que esta tabela não entende de ruas, mas sim de segmentos de ruas. Vou escolher o segmento 58740321 (segmento da Rua do Catete entre a Corrêa Dutra e a Buarque de Macedo):

The screenshot shows a PostgreSQL SQL Editor window. The menu bar includes File, Edit, Query, Favourites, Macros, View, and Help. The toolbar contains icons for file operations, query execution, and help. The SQL Editor tab is active, showing the following query:

```
select
    id, osm_id, osm_name, source, target
from
    public.osm_2po_4pgr
where
    osm_id = 58740321
```

Below the query editor is the Output pane, which is currently displaying the Data Output tab. The results are shown in a table with the following columns: id, osm_id, osm_name, source, and target.

	id integer	osm_id bigint	osm_name character varying	source integer	target integer
1	436230	58740321	Rua do Catete	6454	6450

Agora que achamos os dados do segmento, podemos perguntar quem chega até ele e para onde ele vai. Observe o seu valor *source* 6454. Vamos perguntar: “quem tem como destino o segmento cujo *source* é 6454?” ou, “dado o segmento com origem 6454, quem chega até ele?”

The screenshot shows a PostgreSQL SQL Editor window. The menu bar includes File, Edit, Query, Favourites, Macros, View, and Help. The toolbar contains various icons for file operations, query execution, and help. The SQL Editor tab is active, showing a query in the SQL Editor pane. The query is:

```
select
    id, osm_id, osm_name, source, target
from
    public.osm_2po_4pgr
where
    target = 6454
```

Below the SQL Editor is the Output pane, which is currently showing the Data Output tab. The results are displayed in a table with the following columns: id, osm_id, osm_name, source, and target. The table contains two rows of data:

	id integer	osm_id bigint	osm_name character varying	source integer	target integer
1	4915	5136172	Rua Buarque de Macedo	6453	6454
2	435938	58586064	Rua do Catete	6533	6454

Encontramos duas ruas cujo *target* é a rua com source 6454: o segmento anterior da Rua do Catete (não por coincidência o segmento 58586064), e a Rua Buarque de Macedo.



Ótimo! Descobrimos de onde viemos, mas para onde vamos? Precisamos procurar as ruas com *source* igual ao *target* (6450) do nosso segmento inicial (58740321):

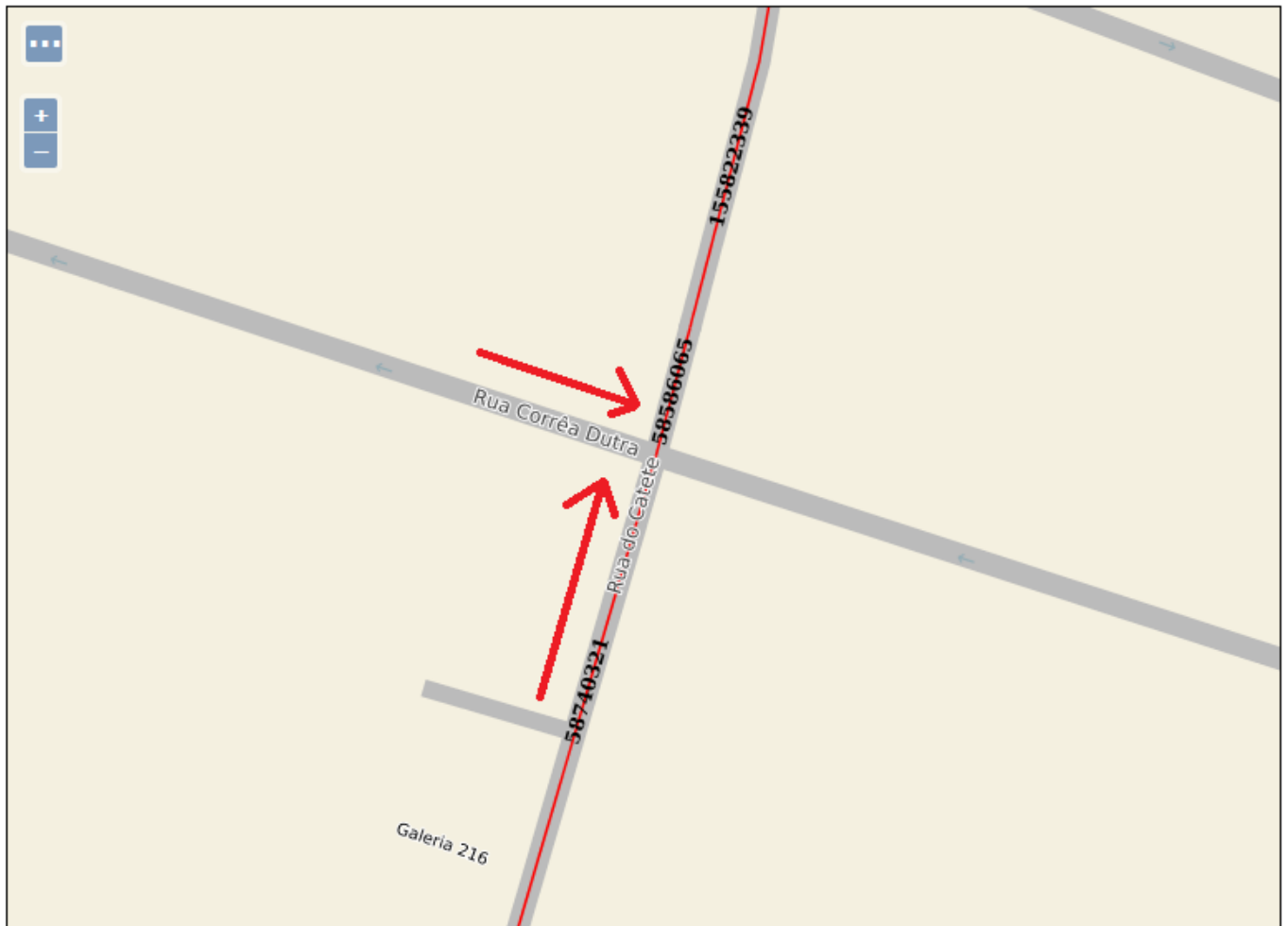
The screenshot shows a SQL Editor window with a menu bar (File, Edit, Query, Favourites, Macros, View, Help) and a toolbar. The 'SQL Editor' tab is active, displaying a query in the 'Previous queries' list. The query is as follows:

```
select
  id, osm_id, osm_name, source, target
from
  public.osm_2po_4pgr
where
  source = 6450
```

Below the query editor is the 'Output pane' with tabs for 'Data Output', 'Explain', 'Messages', and 'History'. The 'Data Output' tab is selected, showing a table with the following data:

	id integer	osm_id bigint	osm_name character varying	source integer	target integer
1	433896	56178677	Rua Corrêa Dutra	6450	267805
2	435939	58586065	Rua do Catete	6450	268919

Encontramos a Rua Corrêa Dutra e outro segmento da Rua do Catete (58586065), mas este segmento não está no mapa! Depois do segmento 58740321 vem o 155822339. Bom, isso é um problema de zoom. O GeoServer “sabe” que precisa omitir algumas coisas para manter a ordem na tela. Então, vamos aproximar um pouco o zoom para ver com mais clareza...



Aí está o segmento perdido bem onde ele deveria estar. Nossa rota então ficou assim: Você pode vir da Rua Buarque de Macedo (segmento 5136172) ou da Rua do Catete (segmento 58586064) , pegar a Rua do Catete (segmento 58740321) e seguir pela Rua Corrêa Dutra (segmento 56178677) ou continuar na Rua do Catete (segmento 58586065) .

Bom, aprendemos a fazer uma rota “na mão”. Na prática, quando você olha para um mapa no Google ou no OpenStreetMap, você não percebe os segmentos de ruas, mas saiba que eles são peças fundamentais no cálculo de rotas. O atributo *osm_id* serve para vincular as topologias de volta para a tabela de dados do OSM *planet_osm_line*, com mais informações sobre o segmento.

Tente agora continuar pela Rua Corrêa Dutra, Praia do Flamengo, Ferreira Viana, retornar pela Rua do Catete, Artur Bernardes e Bento Lisboa.

No próximo artigo, veremos como obedecer a “mão” da rua e começar a preparar o terreno para nosso calculador de rotas.

