



# ConBRepro

XII CONGRESSO BRASILEIRO DE ENGENHARIA DE PRODUÇÃO



## ESG nas Engenharias

30 a 02  
de dezembro 2022

### Proposta de Ferramenta de Importação de Rede Viária do Banco de Dados PostgreSQL/pgRouting para o Simulador SUMO

Elmo Santos da Silva Neto<sup>1</sup>  
Ricardo Villarroel Dávalos<sup>1</sup>

<sup>1</sup>Programa de Pós-Graduação em Engenharia de Transportes e Gestão Territorial  
Universidade Federal de Santa Catarina

**Resumo:** O banco de dados PostgreSQL, por meio das extensões PostGIS e pgRouting, permite armazenamento e manipulação de dados geográficos aliadas à consulta de rotas sobre uma rede viária previamente importada e validada topologicamente. Contudo, mesmo oferecendo tais funcionalidades úteis para a implementação de um modelo de simulação, ainda não dispõe de uma ferramenta que permita o fluxo de dados diretamente para o software de simulação SUMO. O presente trabalho trata do desenvolvimento inicial de uma proposta de integração do banco de dados PostgreSQL ao simulador de mobilidade urbana SUMO, apresentando como estudo de caso uma interseção semaforizada em um dos principais cruzamentos próximo à Universidade Federal de Santa Catarina - UFSC, na cidade de Florianópolis. O trabalho utiliza programação em linguagem Python para consultar o banco de dados e recuperar nós e vértices da área de interesse com o objetivo de construir o arquivo estruturado que é a entrada para a criação de modelos viários no SUMO. A análise de resultados mostra algumas restrições encontradas na implementação e discute-se quais são possíveis soluções a serem investigadas para contorná-las.

**Palavras-chave:** Microssimulação, Modelagem, Mobilidade Urbana, Banco de Dados, Desenvolvimento

### Proposal of Road Network Importing Tool from PostgreSQL/pgRouting Database to SUMO Software

**Abstract:** The PostgreSQL database, through the PostGIS and pgRouting extensions, allows the storage and manipulation of geographic data combined with the query of routes on a previously imported and topologically validated road network. However, even offering such useful functionalities for the implementation of a simulation model, it still does not have a tool that allows the flow of data directly to the SUMO simulation software. The present work is about the initial development of a proposal to integrate the PostgreSQL database to the SUMO urban mobility simulator, presenting as a case study a traffic light intersection at one of the main intersections near the Federal University of Santa Catarina - UFSC, in the city of Florianópolis. The work uses Python programming to query the database and retrieve nodes and vertices from the area of interest in order to build the structured file that is the input for creating road models in SUMO. The analysis of results shows some restrictions found in the implementation and it is discussed which are possible solutions to be investigated to solutionate them.

**Keywords:** Microsimulation, Modeling, Urban mobility, Database, Development

## 1. Introdução

A simulação computacional é uma das ferramentas importantes para a criação e melhoria da base de conhecimento sobre eventos e comportamentos acerca de processos que ocorrem no mundo real. Além de ferramenta, também é um arcabouço teórico que é a base desta atividade, desde a modelagem até a análise de resultados de cada uma das simulações que nortearão tomadas de decisão. As técnicas de simulação utilizam recursos computacionais que tornam mais prática e rápida a atividade de projetar, modelar e implementar simulações. Segundo Law (2015), o aprimoramento de ferramentas computacionais com propósito específico para simulação foi um dos principais fatores que levaram ao aumento de utilização destas ferramentas e consequente aumento da popularidade da simulação em si nos últimos anos.

No contexto de mobilidade urbana, se faz necessária a representação do sistema viário das cidades, com suas pistas, faixas de rolamento, interseções e dados de referência de demanda de tráfego. Como em outros escopos da simulação, o objetivo é simular com maior verossimilhança possível o cenário de estudo, apropriando-se do termo “digital twin”, que segundo Sánchez-Vaquerizo (2022), caracteriza-se pela modelagem e simulação de um sistema que se aproxime tanto quanto possível da realidade, criando uma interação bidirecional entre o ambiente físico e o modelo virtual por meio da representação de entidades relevantes e/ou imprescindíveis ao sistema.

Dado o escopo deste trabalho, será analisada a etapa de modelagem do sistema viário de uma região da cidade de Florianópolis utilizando o software SUMO tendo como base dados vetoriais georreferenciados armazenados em banco de dados relacional do PostgreSQL, que por meio das extensões PostGIS e pgRouting, possibilita armazenamento de dados geográficos e cálculo de rotas sobre estes dados.

A principal motivação do trabalho é a problemática relacionada à criação de redes de infraestrutura viária urbana no SUMO, que apesar de oferecer ferramentas que tornam esta tarefa mais prática, ainda tem limitações como o não-reconhecimento de algumas vias imprescindíveis à modelagem do cenário estudado e impossibilidade de importação de vias de outras fontes de dados cartográficos vetoriais que não a do OpenStreetMap.

Em algum momento, pode ser necessária a importação de dados vetoriais provenientes de fontes como levantamentos de órgãos de mobilidade, universidades ou prefeituras, dessa forma é possível integrar bases utilizando um sistema de informações geográficas (SIG) e importar esta base integrada a um banco de dados que pode ser consultado para importação de redes viárias para o SUMO. Esta importação pode ser realizada em lote, por meio da conversão de tabelas do banco de dados (e seus relacionamentos com outras tabelas) em arquivos estruturados seguindo a sintaxe de definição de elementos de rede neste software de simulação.

Quanto às seções deste trabalho:— A seção 1 contextualiza e introduz o conceito de simulação computacional e o problema abordado pelo trabalho;

— A seção 2 fornece o referencial teórico que fundamentou o desenvolvimento deste trabalho;

— A seção 3 detalha as ferramentas e métodos utilizados no trabalho;

— A seção 4 mostra a metodologia utilizada em cada etapa do trabalho;

— A seção 5 discute os resultados e mostra caminhos interessantes para evolução da pesquisa sobre o tema.

## 2. Referencial Teórico

## 2.1. Microssimulação de tráfego

A microssimulação de tráfego consiste na modelagem virtual da infraestrutura viária e do comportamento de veículos e condutores. Essa modelagem é realizada visando reproduzir as interações entre esses elementos através de diversos algoritmos, dos quais se destacam a perseguição de veículos, troca de faixas e aceitação de brechas. (NORONHA, 2016)

A representação de uma malha viária urbana é realizada através do conceito de rede. Modelos de redes são construídos através da topologia arco-nó. Essa representação nada mais é que um grafo, onde são armazenadas informações sobre recursos de cada trecho ou nó da rede. (SANTOS, 2006)

Como mostrado por Santos (2006), a representação de uma rede viária está ancorada à teoria dos grafos, em que os nós representam interseções, entroncamentos ou qualquer conexão entre vias distintas no cenário real modelado e os arcos são elementos lineares, como ruas, avenidas, alças de entrada ou qualquer outro tipo de via de tráfego que conecte os nós citados anteriormente.

A modelagem da rede é uma etapa importante do processo de microssimulação pois a rede deve ter um nível de detalhamento elevado. A base vetorial utilizada para a modelagem pode ser de fontes variadas, desde OpenStreetMap até levantamentos feitos por entidades especializadas. Dingil et. al (2018) abordam alguns dos problemas de modelagem de rede viária com propósito de simulação a partir de dados do OpenStreetMap:

- incompletude: vias existentes no sistema, mas que ainda não estão presentes na base do OSM;
- imprecisão: interseção entre duas vias que deveria estar, mas não está representada na base do OSM, o que pode acarretar em caminhos “impossíveis” no sistema, quando na realidade são possíveis;
- ambiguidade: descrição de dados de vias podem não dar informações suficientes para a modelagem, como exemplo o número de faixas em vias de mão dupla que pode estar informado na totalidade, não informando a quantidade de faixas em cada sentido;
- erros em geral: identificação mais complexa que podem demandar esforço manual.

## 2.2. OpenStreetMap

O projeto foi criado em 2004 no Reino Unido e segundo Arsanjani et al (2015), a missão do projeto é a criação de um mapa editável do mundo em que os contribuidores e os utilizadores da base cartográfica não são obrigados a produzir ou consumir dados geográficos da maneira convencional oferecida por grandes empresas, tornando esta base livre de restrições licenciasais e/ou autorais.

O OpenStreetMap é disponibilizado em dados abertos, sob a licença Open Data Open Database License (ODbL) pela Fundação OpenStreetMap(OSMF). Tem o direito de copiar, distribuir, transmitir e adaptar os nossos dados, desde que atribua a autoria do OpenStreetMap e os seus contribuidores. Se alterar ou adicionar conteúdo dos nossos dados, pode distribuir o resultado apenas com a mesma licença. (OPENSTREETMAP, 2022)

## 2.3. SUMO

Kurczveil e López (2015) mostram que o desenvolvimento do SUMO foi inicialmente realizado pelo Institute of Transportation Systems of the German Aerospace Center (DLR) em 2001 e evoluiu até o ponto em que se encontra hoje, uma ferramenta de simulação com várias funcionalidades e capaz de representar redes complexas de tráfego. As principais características do software são:

- Código fonte aberto: qualquer pessoa interessada em analisar e modificar o código fonte de componentes do software, tem a liberdade para isso;
- Multiplataforma: funciona em sistemas operacionais como Windows e Linux, necessitando apenas de uma instalação Python para execução de seus módulos;
- Multimodalidade de transportes: além da tradicional modelagem e simulação do fluxo de tráfego de veículos e passeio e de transporte de cargas, também é possível realizar a simulação de transporte público e modos de transporte ativo, como bicicletas e pedestres, dentre outros;
- Simulação microscópica: as entidades simuladas podem ter comportamentos específicos associados, como o comportamento individualizado de motoristas e de classes de veículos, dentre outros.

As ferramentas do SUMO que auxiliam o processo de modelagem do sistema viário também merecem menção, pois tornam um processo moroso e trabalhoso em uma tarefa de alguns minutos, a depender do tamanho da região que deseja-se representar. Para este trabalho, foram utilizados e analisados os resultados destas ferramentas de modelagem do SUMO:

- OSM Web Wizard: utilitário Python que abre uma página na Web para seleção de área de interesse de estudo para posterior download e criação da rede com base nos produtos cartográficos do OpenStreetMap.
- NetEdit: utilitário Python que fornece uma interface gráfica para edição de vias e suas conexões, permitindo a alteração intersecções, remoção de movimentos proibidos, adição de semáforos e demais informações pertinentes à modelagem do fluxo de entidades pelo sistema.
- netconverter: utilitário de linha de comando que converte em arquivo .net.xml que contém as definições de uma rede válida no SUMO os arquivos de entrada contendo especificações de posição e identificação de nós, vértices e conexões da rede.

### **2.3. Python**

Python é uma linguagem de programação multiparadigma criada por Guido van Rossum em 1993. Segundo Pérez et al (2010), a sua ampla utilização no meio acadêmico se deve ao desenvolvimento de ferramentas de código aberto capazes de auxiliar em novos desafios de pesquisa. Existem bibliotecas de propósito geral que auxiliam na modelagem de problemas comuns a mais de uma área de conhecimento e também existem bibliotecas de propósito específico, no caso da simulação, há o exemplo da biblioteca SimPy. A linguagem Python, junto com Java e C++, é uma das bases do software SUMO e muitos dos seus utilitários de linha de comando foram escritos nela. Por este motivo, a solução proposta por este trabalho foi desenvolvida em Python.

### **2.4. pgRouting**

O pgRouting é uma extensão do banco de dados PostgreSQL que trabalha em conjunto com a extensão PostGIS. O PostgreSQL é um sistema gerenciador de banco de dados criado em 1986 pela Universidade de Berkeley. (POSTGRES, 2022). O PostgreSQL é frequentemente citado em pesquisas como a Stack Overflow, como um dos sistemas gerenciadores de banco de dados mais utilizados por sua comunidade de desenvolvedores e pesquisadores. (STACKOVERFLOW, 2021)

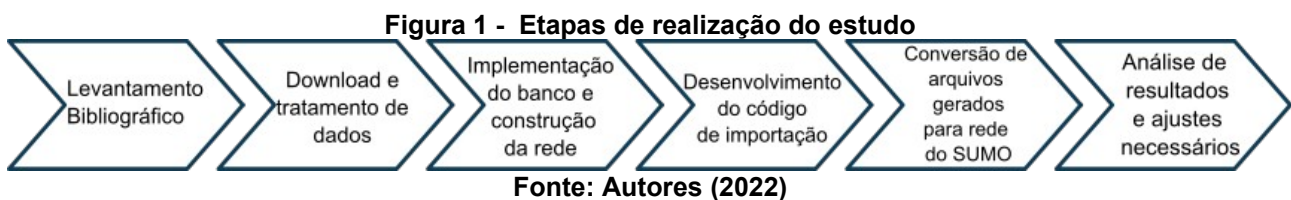
O PostGIS é uma extensão que provê ao banco de dados a capacidade de armazenamento, manipulação e consulta de dados georreferenciados que podem ser importados/exportados para Sistemas de Informações Geográficas, como QGIS. (GRASER; DRAGASCHNIG, 2020)

Segundo Choosumrong et. al (2012), a extensão pgRouting estende o banco de dados PostgreSQL/PostGIS para prover funções de roteamento geoespacializadas. Esta extensão depende da extensão PostGIS para seu funcionamento e adiciona ao banco de dados um conjunto de funções para criação de redes (ou grafos), análise de possíveis erros de topologia, navegação e outras operações concernentes ao estudo de grafos. (PGROUTING, 2022)

### 3. Materiais e Métodos

O procedimento para o desenvolvimento deste trabalho pode ser considerado um estudo de caso, definido como uma análise de natureza empírica que investiga um determinado fenômeno, geralmente contemporâneo, dentro de um contexto real de vida, quando as fronteiras entre o fenômeno e o contexto em que ele se insere não são claramente definidas (MIGUEL, 2007).

A figura 1 retrata as etapas que serviram de apoio na realização deste estudo. A primeira etapa foi a busca de materiais, manuais e publicações que fortaleceram a ideia. Depois foram realizadas as capturas dos vídeos, com levantamento e modelagem dos dados, posteriormente teve foco a análise e correlação dos resultados, bem como a conclusão deste estudo.



A estrutura metodológica está organizada conforme exposto a seguir, com uma explicação de como os dados foram obtidos e trabalhados levando em conta as dificuldades observadas.

— Na etapa 1, o levantamento bibliográfico foi realizado por meio de consulta a trabalhos correlatos e relevantes da área e também por consulta a documentação de cada uma das ferramentas utilizadas, principalmente o SUMO, pois é necessário compreender em detalhes a estrutura de cada arquivo de entrada para a criação de redes no SUMO;

— As etapas 2 e 3 são realizadas foram realizadas em conjunto para este trabalho. O download de dados foi realizado a partir do OpenStreetMap, por meio de código criado pelos próprios autores, que busca todos os dados vetoriais do sistema viário na extensão de uma cidade e realiza validações topológicas na rede por meio de funções do pgRouting;

— Na etapa 4, foi realizado o desenvolvimento de script em Python que busca, a partir da delimitação de uma área de interesse dentro da cidade, as informações sobre arcos e nós da rede no banco de dados criado na etapa anterior. Com estes dados, o script tem o necessário para a criação dos arquivos descritores de topologia de rede que serão convertidos em rede do SUMO;

— Na etapa 5, é realizada a conversão dos arquivos gerados pelo script da etapa anterior em uma rede do SUMO. Arquivos auxiliares para geração de viagens aleatórias e rotas associadas a elas também foram criados para formar um conjunto de arquivos básico de simulação e passível de execução dentro do software;

— Na etapa 6, os resultados gerados são analisados e alguns ajustes são feitos na rede, tendo em vista as restrições encontradas durante a implementação da metodologia.

## 4. Estudo de Caso

A região escolhida como estudo de caso para a aplicação do método proposto foi o cruzamento da Rua Deputado Antônio Edu Vieira, com 2 sentidos segregados de 3 faixas cada, com outra duas ruas:

a) Rua João Pio Duarte Silva, de mão dupla, com 2 faixas em direção à Rua Deputado Antônio Edu Vieira e 1 faixa em sentido oposto em direção à região leste da Ilha de Florianópolis

b) Rua Delfino Conti, de mão dupla, com uma faixa em direção à Rua Deputado Antônio Edu Vieira e outra em sentido oposto, em direção à oeste (centro) da cidade.

A localização deste cruzamento pode ser visualizada no mapa abaixo.

**Figura 2 – Interseção estudada**



**Fonte: Adaptado de OpenStreetMap (2022)**

### 4.1. Dados de Entrada

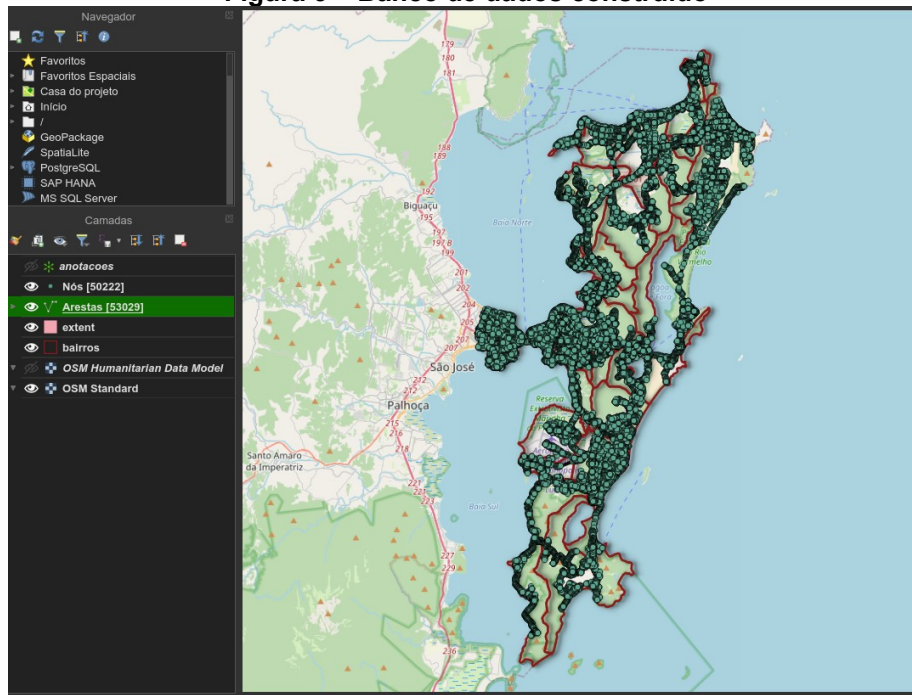
O banco de dados utilizado neste trabalho foi construído por meio de um script escrito em Python que faz o download do sistema viário de uma cidade específica no OpenStreetMap e realiza as validações topológicas necessárias para que não existam nós conectados por um arco no sistema real, mas que na representação estejam inacessíveis por conta de uma modelagem descuidada em relação a estes aspectos importantes que asseguram a construção de uma rede de qualidade. Não é do escopo deste trabalho a criação deste banco, que como citado anteriormente, pode agregar várias fontes de dados vetoriais. Contudo, para melhor entendimento da estrutura das tabelas do banco, as etapas de construção do banco são brevemente descritas:

- Criação de um banco vazio no PostgreSQL com as extensões PostGIS e pgRouting habilitadas;
- Download do sistema viário para veículos automotores na cidade de Florianópolis a partir do OpenStreetMap;
- Segmentação de vias para assegurar topologia correta da rede;
- Importação dos dados vetoriais para o banco de dados.
- Criação da nós e conexão entre nós na rede, com preenchimento de campos de origem e destino em tabelas que armazenam os arcos (vias) da rede e preenchimento dos custos associados a cada arco (utilizadas somente a distância e velocidade permitida)

É importante citar que o banco foi criado já com a configuração de um sistema de coordenadas, projetado e em unidade métrica, apropriado para o Estado de Santa Catarina e utilizado como padrão pela América Latina: o SIRGAS UTM Zona 22 S.

O banco utilizado tem mais de 50 mil vértices e mais de 50 mil arestas, como mostra a Figura 2, o que o torna um caso de estudo interessante para testes da metodologia em áreas maiores, que vão aumentando gradativamente a área de abrangência.

**Figura 3 – Banco de dados construído**



**Fonte: Autores (2022)**

## 4.2. Desenvolvimento do script

O script foi desenvolvido em Python e depende basicamente de apenas uma biblioteca, o que é interessante para manter o baixo número de dependências e tornar a manutenção no código mais simples.

O script inicia com a conexão ao banco de dados, fornecendo o conjunto padrão de informações que darão acesso às tabelas e seus registros que a base contém. Após a conexão, é feita a intersecção de um polígono (informado pelo usuário do script) com os nós presentes na tabela "vértices" do banco. Uma vantagem em relação ao OSM Web Wizard, é que este utilitário disponível na instalação padrão do SUMO, permite a seleção de regiões retangulares, enquanto a ferramentas propostas permite a seleção de regiões retangulares por meio da digitação de coordenadas dos extremos deste retângulo e também por polígonos irregulares fornecidos pelo usuário em um arquivo GeoJSON.

Logo após a busca de vértices ser feita, eles são armazenados em um array declarado no script e em sequência todos os arcos (arestas ou vias) do banco que referenciam esses vértices como origem ou destino também são buscados e armazenados em um array dentro do script. Cada uma destas arestas contém informações sobre a estrutura da via, como a superfície, a velocidade permitida, o número de faixas etc. Um laço de repetição é executada para percorrer este array e construir dois arquivos importantes para a modelagem do sistema viário: o arquivo \*.edg.xml e o arquivo \*.nod.xml.

O arquivo \*.nod.xml tem a seguinte estrutura, segundo arquivo \*.xsd que define um conjunto de nós de uma rede:

Figura 4 – Estrutura do arquivo \*.nod.xml

```
1 <nodes>
2 <node id="1" x="longitude" y="latitude" type="priority" [opcionais]>
3 <node id="2" x="longitude" y="latitude" type="traffic_lights" [opcionais]>
4 ...
5 </nodes>
```

Fonte: Autores (2022)

O arquivo \*.edg.xml tem a seguinte estrutura, segundo arquivo XSD que define um conjunto de nós de uma rede:

Figura 5 – Estrutura do arquivo \*.edg.xml

```
1 <edges>
2 <edge id="1" from="1" to="1" [opcionais]>
3 <edge id="2" from="2" to="3" [opcionais]>
4 <edge id="3" from="3" to="2" [opcionais]>
5 ...
6 </edges>
```

Fonte: Autores (2022)

Alguns pontos importantes sobre a criação dos arquivos de entrada:

a) O atributo “oneway” de cada uma das arestas é considerado na criação de arcos. Se a via é oneway, apenas um arco é criado do nó source para o nó target. Se não é oneway, são criados dois arcos: um saindo do nó source para o nó target e outro do nó target para o nó source. Foi uma escolha de implementação usar dois arcos e não a definição “neigh lanes” por se mostrar uma escolha mais segura para o objetivo alcançado em testes iniciais

b) Os vértices têm um atributo indicando se são “traffic\_lights”. Se são, o atributo priority é marcado como “traffic\_lights”, sem definição do tipo de semáforo para tornar a implementação mais simplificada. Também não foram criados planos semaforicos automaticamente por conta de ainda não haver uma representação bem definida dessa informação nas tabelas de um banco típico com PostGIS e pgRouting habilitados

Depois dos arquivos gerados, o utilitário de linha de comando netconverter é executado passando como argumento os arquivos de entrada e o arquivo de saída que conterà as especificidades da rede interpretada a partir destes arquivos.

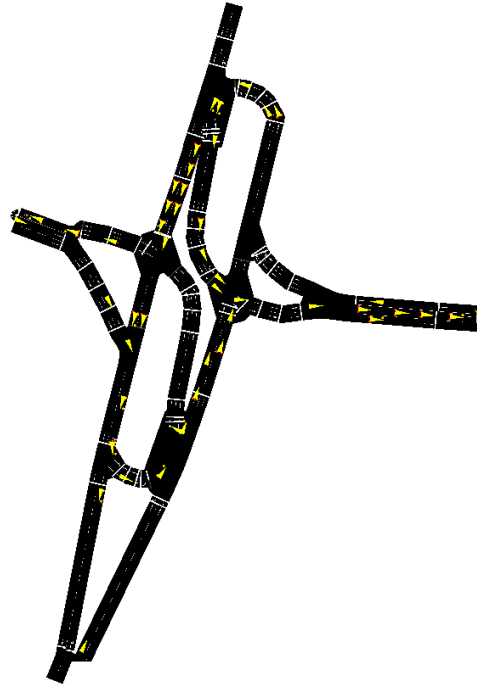
Com o objetivo de realizar simulações de testes e observar o comportamento do sistema, foram gerados dados de viagens aleatórias com o utilitário randomTrips. As rotas entre os pontos de origem e destino das viagens foram calculadas com o utilitário duarouter. É importante frisar que o objetivo era observar a corretude da rede gerada pela metodologia proposta e não a análise de resultados quantitativos dos produtos da simulação. Contudo, essa é a base para a construção de sistemas reais que se aproximem mais da realidade de campo, utilizando, por exemplo, a contagem de veículos em horários de pico.

### 4.3 Resultados

A rede gerada pela metodologia e com uma simulação sendo executada pode ser vista na Figura 6.



**Figura 6 – Rede gerada pelo netconverter**



**Fonte: Próprio autor (2022)**

Pôde ser observado que todas os nós e arestas definidos nos arquivos de entrada foram respeitados, o que sugere uma robustez na criação de redes maiores, pois o script não está dependente de instabilidades na conexão do computador com a Internet ou de memória suficiente no navegador que está hospedando o utilitário Web Wizard.

O número de faixas em cada sentido também foi criado corretamente, observando principalmente a Rua Pio que tem um número diferente de faixas em cada sentido. Os semáforos também foram importados, mas não geraram o resultado esperado, o que sugere um estudo mais detalhado do arquivo de rede gerado pelo netconvert para inserção da semaforização dentro deste próprio arquivo. As junções entre vias foram bem estabelecidas, exceto na rua citada anteriormente, em que o número diferente de faixas em cada sentido deslocou a geometria original da via. Os demais nós e arestas se mantiveram próximos à geometria original, mantendo-se inclusive tão precisos quanto os dados originais do OpenStreetMap. Assim, há também uma referência geoespacial bem estabelecida caso seja necessário, por exemplo, importar uma imagem de satélite para utilizar em segundo plano visual na simulação.

Movimentos proibidos na realidade de campo e que foram permitidos na simulação, por falta de uma estrutura bem definida de mapeamento destas restrições. A extensão pgRouting tem estruturas de dados e funções em caráter experimental para mapeamento de restrições de manobra, porém por uma escolha de implementação, as restrições de manobra foram mapeadas retirando-se manualmente as conexões proibidas entre arcos, como exemplificado na Figura 4. Obviamente, isto se torna inviável para redes maiores e é necessário buscar uma alternativa para automatizar este mapeamento.

Figura 4 – Edição de rede no utilitário netedit do SUMO



Fonte: Próprio autor (2022)

## 5. Conclusões

O trabalho buscou implementar a proposta de uma metodologia inicial para integração do banco de dados PostgreSQL com as extensões PostGIS e pgRouting ao software SUMO, dando mais flexibilidade ao pesquisador/profissional da área de transportes realizar a simulação de sistemas de transportes com dados provenientes de fontes diversas, convergindo para uma base de dados única. Pelo fato de serem ferramentas com uma base de usuários relevante, e por não haver ainda na literatura nem em repositórios de software livre alguma ferramenta que faça a integração entre elas, o trabalho buscou a originalidade da implementação.

A proposta foi desenvolvida com base nas documentações oficiais das ferramentas utilizadas e também em trabalhos similares que dão foco à modelagem de redes. A linguagem Python foi a escolha mais natural para a implementação do trabalho dada a considerável integração de suas várias bibliotecas de propósito científico com outras ferramentas já utilizadas na pesquisa. O SUMO tem vários de seus utilitários escritos em Python, o que também torna a integração mais simplificada. O script em Python fez o intermédio entre o banco de dados e o SUMO, gerando a partir de consultas a tabelas do banco os arquivos XML que alimentaram o utilitário de criação de redes do SUMO.

Algumas limitações foram encontradas, como a impossibilidade de criação automática de um plano semafórico, pois esta informação não está disponível no OpenStreetMap nem foi encontrada uma estrutura bem definida para especificação desta configuração. Ainda é necessário encontrar alternativas para mapeamento de restrições de manobra na rede, já que para áreas de abrangência maiores em simulações futuras, é inviável a realização deste trabalho de forma manual.

Para trabalhos futuros, pretende-se explorar mais a fundo as dificuldades encontradas e citadas anteriormente e sobretudo, explorar a importação do sistema viário para outros modos de transporte e também de informações sobre o transporte público. Dessa forma, com uma ferramenta mais robusta, é possível publicar esta ferramenta para contribuição de outros pesquisadores e possivelmente integrá-la à distribuição oficial do SUMO.

## Referências

- ARSANJANI, J. J. et al. An introduction to OpenStreetMap in Geographic Information Science: Experiences, research, and applications. **OpenStreetMap in GIScience**. Springer, Cham, 2015. p. 1-15.
- CHOOSUMRONG, S.; RAGHAVAN, V.; BOZON, N. Multi-criteria emergency route planning based on analytical hierarchy process and pgRouting. **Geoinformatics**, v. 23, n. 4, p. 159-167, 2012.
- DINGIL, A. E. et al. Road network extraction with OSMNx and SUMOPy. **EPiC Series in Engineering**, v. 2, p. 111-117, 2018.
- GRASER, A.; DRAGASCHNIG, M. Open geospatial tools for movement data exploration. **KN-Journal of Cartography and Geographic Information**, v. 70, n. 1, p. 3-10, 2020.
- KURCZVEIL, T.; LÓPEZ, P. A. eNetEditor: Rapid prototyping urban traffic scenarios for SUMO and evaluating their energy consumption. In: SUMO 2015 – Intermodal Simulation for Intermodal Transport, Berlim. **Proceedings**. Berlim: Deutsches Zentrum für Luft-und Raumfahrt, 2015. p 137-160.
- LAW, A. M. **Simulation Modeling and Analysis**. Nova Iorque: McGraw-Hill Education, 2015.
- MIGUEL, P. A. C. Estudo de caso na engenharia de produção: estruturação e recomendações para sua condução. **Production**, v. 17, n. 1, p. 216-229, 2007
- NORONHA, H. D. B. **Estudo comparativo de dois microssimuladores de tráfego - AIMSUN e VISSIM**. Florianópolis - SC, 144 p., 2016. Trabalho de Conclusão de Curso (Engenharia Civil). Universidade Federal de Santa Catarina.
- OPENSTREETMAP. **OpenStreetMap**. Disponível em: <<https://www.openstreetmap.org/copyright>> Acesso em: 22 set. 2022.
- PEREZ, F.; GRANGER, B. E.; HUNTER, J. D. Python: an ecosystem for scientific computing. **Computing in Science & Engineering**, v. 13, n. 2, p. 13-21, 2010.
- PGROUTING. **pgRouting**. Disponível em: <<https://pgrouting.org/>> Acesso em: 22 set. 2022.
- SÁNCHEZ-VAQUERIZO, J. A. Getting Real: The Challenge of Building and Validating a Large-Scale Digital Twin of Barcelona's Traffic with Empirical Data. **ISPRS International Journal of Geo-Information**, v. 11, n. 1, p. 24, 2022.
- SANTOS, J. A. **Modelagem de Malhas Viárias Urbanas Aplicando Conceitos de Grafos**. Dourados – MG, 68 p., 2006. Trabalho de Conclusão de Curso (Ciência da Computação). Universidade Estadual de Mato Grosso do Sul.
- STACKOVERFLOW. **2021 Developer Survey**. Disponível em: <<https://insights.stackoverflow.com/survey/2021>> Acesso em: 22 set. 2022.