Links úteis para a simulação:

- Repositório no GitHub com todos os materiais: (precisa de permissão para acessar)
  https://github.com/santiagocardoso/ic-santiagocardoso

- Documentação do NS3:
  https://www.nsnam.org/docs/release/3.29/tutorial/ns-3-tutorial.pdf

- Instalação do MINUET:
  https://bitbucket.org/lumo-ufpb/sociable/src/7ce75910c91e7de50ce590bd86ad9967d0e099e3/ns-3.28/src/minuet/?at=develop-minuet
- Documentação SUMO:
  https://sumo.dlr.de/docs/Installing/Linux_Build.html

- Documentação NS3: https://www.nsnam.org/releases/ns-3-29/documentation/
- Documentação NS3 Legacy: https://www.nsnam.org/documentation/older/
- Classes NS3: https://www.nsnam.org/doxygen/classes.html
- Docker Compose: https://docs.docker.com/compose/

- Módulo de avaliação (ATENÇÃO: não alterar o repositório a pedido do Alisson Yuri):
  https://drive.google.com/drive/folders/1Ib2QBK-fTFVHBd6grBW8CdTEW47ElINc?usp=sharing

Links importantes

Disponibilizados pelo Laboratório de pesquisa: Lumo / Autor: Carlos Augusto

- Tutorial do LuST + SUMO:
  https://bitbucket.org/lumo-ufpb/sociable/src/7ce75910c91e7de50ce590bd86ad9967d0e099e3/ns-3.28/src/minuet/doc/scenarioTrace.md
- Tutorial do MINUET:
  https://bitbucket.org/lumo-ufpb/sociable/src/7ce75910c91e7de50ce590bd86ad9967d0e099e3/ns-3.28/src/minuet/?at=develop-minuet
- Manual do Trace de Vídeo:
  https://bitbucket.org/lumo-ufpb/sociable/src/7ce75910c91e7de50ce590bd86ad9967d0e099e3/ns-3.28/src/minuet/doc/video-stream.md

Documentação ns-3.29: https://www.nsnam.org/releases/ns-3-29/documentation/

Instalação local do ambiente para realizar as simulações

Vá para o local que você deseja instalar o simulador, eu criei uma pasta na minha home do Ubuntu ~/projects/

```
$ cd ~/projects/
$ mkdir ic
$ cd ic
$ wget https://www.nsnam.org/release/ns-allinone-3.29.tar.bz2
```

```
$ tar xjf ns-allinone-3.29.tar.bz2
```

Para instalar o MINUET pegue do repositório do GitHub https://github.com/santiagocardoso/ic-gradis-sociable-fork.git e coloque esse módulo do MINUET em ~/ns-allinone-3.29/ns-3.29/src/ após isso somente precisamos realizar o build da simulação.

```
$ cd ns-allinone-3.29
$ ./build.py --enable-examples --enable-tests
```

Caso ocorra um erro com o Python você terá que abrir esses arquivos .py e na primeira linha onde tem algo parecido com “#! /usr/bin/env python” reescreva para “#! /usr/bin/env python3”, ou seja, somente adicione o 3 da nova versão do python no final, são poucos os arquivos que terão que ser atualizados dessa forma.

Teste se foi instalado corretamente com:

```
$ ./test.py
```

Irá aparecer algo parecido com isso:

92 of 92 tests passed (92 passed, 0 failed, 0 crashed, 0 valgrind errors)

O importante é que todos os testes passem, não tem problema caso alguns sejam pulados, caso ocorra algum erro ou ocorra um crash precisa ser verificado o problema.

Para executar a simulação crie um arquivo “auto.bash” e cole esse trecho de código:

```
#!/bin/bash

current_dir=$(pwd)

cd ../../
CXXFLAGS='-Wall -g -O0' ./waf configure --build-profile=debug --enable-examples --enable-tests

./waf --run minuet-scenario
```

Após isso dê permissões de execução:

```
$ chmod +x auto.bash
```

E execute a simulação pelo arquivo toda vez:

```
$ ./auto.bash
```

Talvez ocorra um erro com o path dos arquivos de trace, abra pelo vscode a pasta com o MINUET e abra o arquivo “minuet-utils.cc”, aperte Ctrl+F e escreva “/ns-allinone-3.29/ns-3.29/src/”, aperte alt+enter e atualize todas as linhas para o local que estão seus arquivos de trace.

Seguindo esses passos é para a simulação já estar funcionando corretamente de forma local em seu computador

Agora, para instalar o SUMO precisamos executar alguns comandos:

```
$ cd ~/projects/ic/
```

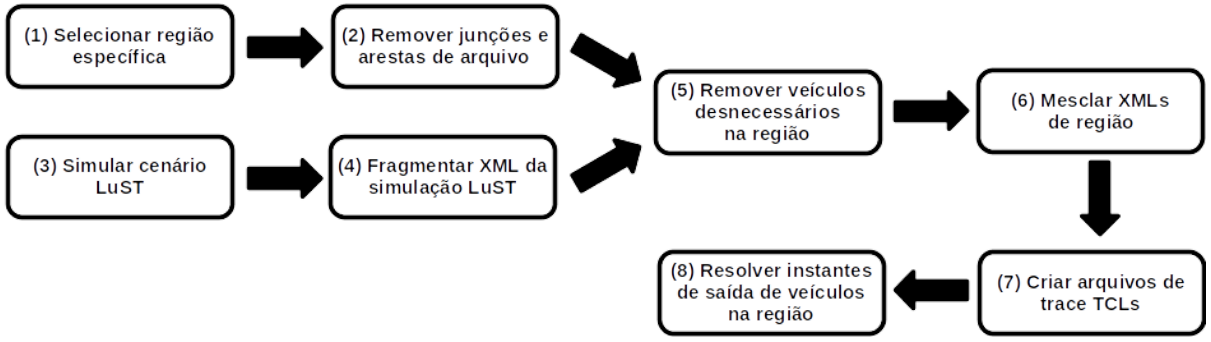
```
$ sudo apt-get install git cmake python3 g++ libxerces-c-dev libfox-1.6-dev libgdal-dev libproj-dev libgl2ps-dev python3-dev swig default-jdk maven libeigen3-dev
$ git clone --recursive https://github.com/eclipse-sumo/sumo
$ cd sumo
$ export SUMO_HOME="$PWD"
$ cmake -B build .
$ cmake --build build -j$(nproc)
```

Com isso é para o SUMO estar funcionando corretamente em seu computador.

Geração dos arquivos de Trace e simulação

Pré requisitos:
SUMO: https://sumo.dlr.de/docs/Installing/Linux_Build.html
LuST: https://github.com/lcodeca/LuSTScenario
MINUET: https://bitbucket.org/lumo-ufpb/sociable/src/7ce75910c91e7de50ce590bd86ad9967d0e099e3/ns-3.28/src/minuet/?at=develop-minuet

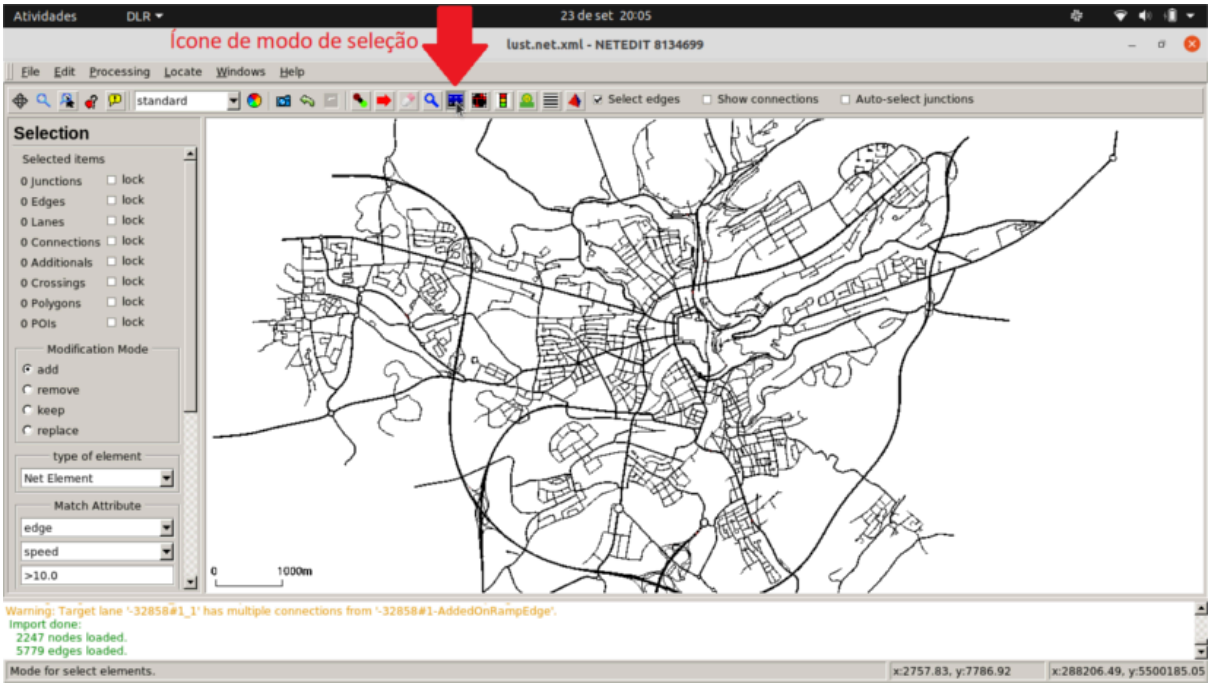
Passos para realizar o trace:



(1) Selecionar e recortar uma área de interesse do cenário LuST para a simulação do MINUET. Inicialmente, execute o editor de malha viário do SUMO *netedit* (localizado em `[caminho_SUMO]/bin/`) pelo comando:

```
$ ./[caminho_SUMO]/bin/netedit
```

Em seguida, pelo *netedit*, abra o arquivo *lust.net.xml* (localizado em `[caminho_LuST]/scenario/lust.net.xml`), o qual define a malha viária do cenário LuST. Após abrí-lo, é possível visualizar a malha viária completa da cidade de Luxemburgo. Para selecionar as vias do cenário é necessário configurar o netedit em modo de seleção, como apresenta a imagem abaixo:

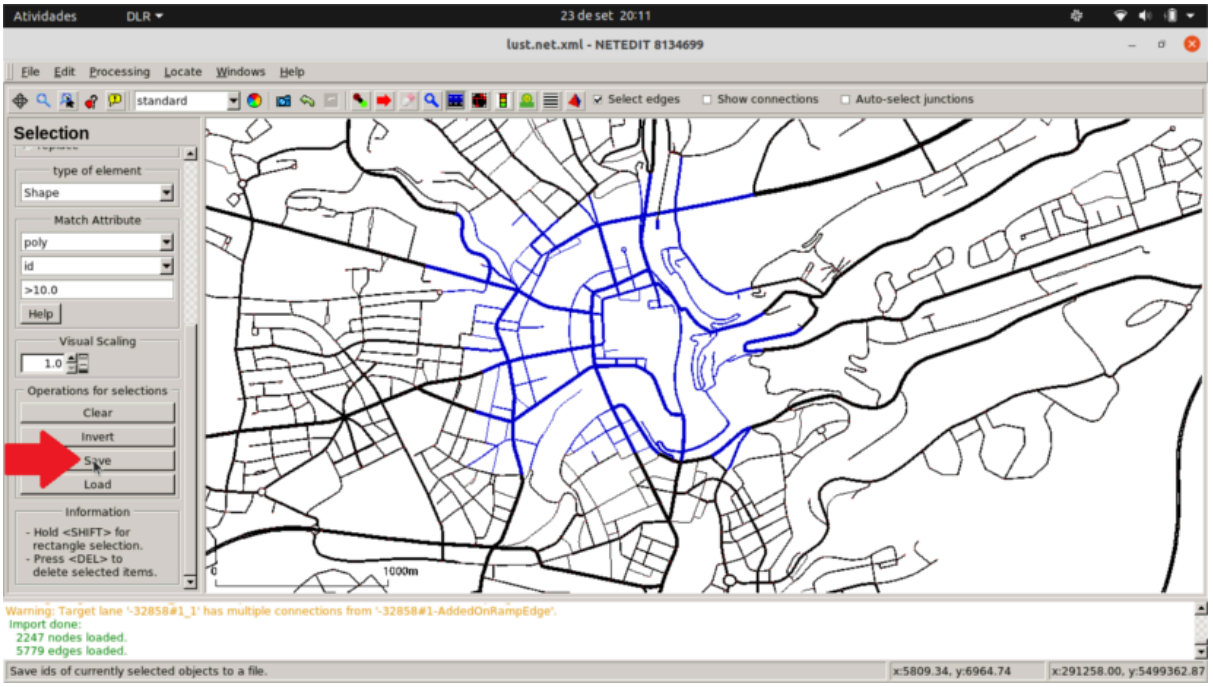


Para selecionar múltiplas vias no cenário, pressione *shift* enquanto seleciona a área específica. Em seguida, salve a região selecionada em um arquivo de texto, clicando no botão *save* (localizado no canto inferior esquerdo), como apresenta a imagem abaixo:

Eu **recomendo selecionar uma área MUITO pequena**, pois a simulação leva um tempo considerável para ser preparada com áreas maiores.

Também tome **cuidado com o armazenamento** de onde está seu repositório, pois nos passos **(3)** e **(4)** ficou executando por ~12h, geraram 30GB de arquivo .xml, e eu já não tinha mais espaço de armazenamento na minha /home então reiniciei esse processo no meu SSD com 512GB.

Recomendo baixar esses arquivos pela pasta no drive que a antiga bolsista proporcionou: https://drive.google.com/drive/folders/1d8KyMPs3d-RhYyWbADxCqFMStcJyTKmR?usp=share_link
Caso decida ir por esse caminho pode pular os passos **(3)** e **(4)**, caso contrário, continue seguindo eles.



Em seguida, para o procedimento **(2)**, abra o arquivo de texto, remova as linhas que contenham o trecho "junction:", remova o trecho "edge:" das linhas que restaram e salve o arquivo. O procedimento **(3)** consiste em simular o cenário LuST completo, pelo SUMO, a fim de gerar o arquivo de trace de saída (esse arquivo é gerado no formato .xml). Para isso, o seguinte comando deve ser executado após a escolha de uma das quatro opções de configurações:

LuST Scenario can be launched directly with four configuration files.

- Mobility: shortest path with rerouting.
  - sumo -c dua.static.sumocfg with static traffic lights.
  - sumo -c dua.actuated.sumocfg with actuated traffic lights.
- Mobility: Dynamic user equilibrium.
  - sumo -c due.static.sumocfg with static traffic lights.
  - sumo -c due.actuated.sumocfg with actuated traffic lights.

```
$ ./[caminho_SUMO]/bin/sumo -c [arquivo_de_configuracao].cfg --fcd-output [LustTrace].xml
```

O meu ficou assim para realizar esse trace já estando dentro da pasta do sumo:

```
$ ./sumo -c ~/projects/ic/LuSTScenario/scenario/due.static.sumocfg --fcd-output "/media/santiago/UBUNTU\ SSD/trace_regiao_central.xml"
```

No procedimento **(4)**, em virtude do grande tamanho do arquivo XML gerado no procedimento anterior, é necessário fragmentá-lo em arquivos XML menores. Para isso, utilizamos o *script* `Split_Trace.py` (localizado em `minuet/utils/scenarios/Luxembourg/scripts`). Antes de rodar este *script*, os parâmetros de configuração devem ser definidos, indicando o caminho de entrada do arquivo de trace original (*PATH_XML_FILE*) e o caminho de saída dos arquivos fragmentados (*PATH_NEW_XML_FILE*). Em virtude do alto custo de tempo e

processamento demandado pelos procedimentos **(3)** e **(4)** já disponibilizamos a saída do procedimento **(4)** em link.

No procedimento **(5)**, removemos os veículos que não pertencem à região do cenário selecionado. Para isso, utilizamos o *script* `Removes_Vehicles_From_the_Trace.py` (localizado em `minuet/utils/scenarios/Luxembourg/scripts`). Este *script* recebe como entrada os arquivos XMLs fragmentados do procedimento **(4)** (*PATH_FILE_TRACING*) e o arquivo da malha viária (.txt) do procedimento **(2)** (*PATH_EDGES*). Também, deve-se definir no *script*, o caminho de saída dos arquivos gerados em *PATH_NEW_FILE_TRACING*.

Para executar ao abrir o arquivo `Removes_Vehicles_From_the_Trace.py` troque "#!/usr/bin/python" por "#!/usr/bin/python2" para poder executar normalmente, e também atualize o número de arquivos de 128 para 211.

Por exemplo o meu ficou dessa forma:

```
PATH_EDGES = "/home/santiago/projects/ic/LuSTScenario/scenario/regiao_central.txt"
PATH_FILE_TRACING = "/media/santiago/UBUNTU SSD/fragmentos/"
PATH_NEW_FILE_TRACING = "/media/santiago/UBUNTU SSD/veiculos_removidos/"
TOTAL_FILES = 211
```

O procedimento **(6)** consiste em mesclar os arquivos de saída do procedimento **(5)** por meio do *script* `Merge_Traces.py`,localizado em `minuet/utils/scenarios/Luxembourg/scripts`. Neste script, defina como entrada os arquivos XMLs, gerados no procedimento anterior, no parâmetro *PATH_XML_FILE* e como caminho do arquivo de saída (i.e. arquivo mesclado) no parâmetro *PATH_NEW_XML_FILE*.

Para utilizar os traces de mobilidade da região selecionada LuST no NS3, é necessário gerar os arquivos TCLs (mobility, activity, config) da região. O arquivo de mobilidade (mobility), informa as posições de cada nó da simulação em função do tempo, o arquivo de atividade (activity), informa os instantes de entrada e saída de cada nó no cenário, e o arquivo de configuração (config), contém as informações de total de nós, tempo e área da simulação. O procedimento **(7)** é responsável por gerar tais arquivos de trace por meio do do *script* traceExporter.py, localizado em `[caminho_SUMO]/tools/`. Para isso, execute os seguintes comandos:

```
$ python2 traceExporter.py --ignore-gaps --fcd-input=[arquivoTracerFinal].xml --ns2mobility-output=mobiTracerFinal.tcl
$ python2 traceExporter.py --ignore-gaps --fcd-input=[arquivoTracerFinal].xml --ns2activity-output=activityTracerFinal.tcl
$ python2 traceExporter.py --ignore-gaps --fcd-input=[arquivoTracerFinal].xml --ns2config-output=configTracerFinal.tcl
```

Por fim, o procedimento **(8)** corrige os tempos de saída de cada veículo no arquivo TCL de atividade (activity). Para tal, utilize o *script* `Define_Start_and_Final_Time_in_Activity_File.py` localizado em

`minuet/utils/scenarios/Luxembourg/scripts`. Esse script recebe como entrada o caminho do arquivo TCL de mobilidade (em *pathMobilityTclFile*) e o caminho do arquivo TCL de atividade (em *pathActivityTclFile*) gerados no procedimento anterior, além do número total de nós (em *totalNodes*), esse número você pode conseguir através do arquivo gerado anteriormente configTracerFinal.tcl. Como saída, o script gera o próprio arquivo TCL de atividade corrigido.

Mover mobiTracerFinal.tcl, activityTracerFinal.tcl, configTracerFinal.tcl para minuet/utils/trace/tcl

Volte à tela do netedit.

De acordo com as coordenadas da seleção, criar arquivos TraceBaseStations.bs, adicionando base stations com coordenadas próximas à seleção no netedit. ID_ESTACAO_BASE POSICAO_X POSICAO_Y

Mover o arquivo TraceBaseStations.bs para minuet/utils/trace/base_station

De acordo com as coordenadas da seleção, criar arquivos TraceEvents.ev, adicionando eventos com coordenadas próximas à seleção no netedit.
ID_EVENTO TEMPO_INICIO EVENTO_E_FIXO POSICAO_X POSICAO_Y POSICAO_Z DURACAO_EVENTO NODE_EVENT

Mover o arquivo TraceEvents.ev para minuet/utils/trace/events.

Gerar os arquivos de trace de vídeo de acordo com o tutorial, porém como os arquivos nunca mudam neste tipo de simulação podem ser baixados aqui: https://drive.google.com/drive/folders/1-jqteVwS5DljgonIYKUCu-7vIQ_DFnDi?usp=share_link

Mover os arquivos de trace de vídeo (.yuv, .mp4 e .trace) para minuet/utils/trace/video_stream/

Abrir o arquivo minuet/model/minuet-utils.h

Modificar FLOOR_SIZE_X, FLOOR_SIZE_Y, FLOOR_SIZE_X_MIN, FLOOR_SIZE_Y_MIN, TOTAL_NODES, TOTAL_TIME_SIMULATION, START_TIME_APP, FINAL_TIME_APP. Estas informações estão no arquivo TCL config gerado nos passos anteriores.

Abrir o arquivo minuet/model/minuet-utils.cc

Modificar: TRACE_MOBILITY_FILE, TRACE_ACTIVITY_FILE, TRACE_CONFIG_FILE, TRACE_EVENTS_FILE, TRACE_BASE_STATIONS_FILE

Modificar as informações de basestation se necessário.

Caso tenha tido algum problema ou dificuldade com a instalação, realização do trace, entre outros, entre em contato com o coordenador da bolsa para pedir ajuda ao bolsista que criou o material.