

# DESENVOLVIMENTO DE PLUGIN PARA GENERALIZAÇÃO SEMI-AUTOMÁTICA DE AMBIENTES INDOOR

Luiz Otávio Carneiro Filho<sup>1</sup>  
Luciene Stamato Delazari<sup>2,3</sup>  
Amanda Pereira Antunes<sup>2,3</sup>  
Marciano da Costa Lima<sup>2,3</sup>  
Leonardo Ercolin Filho<sup>3</sup>

<sup>1</sup> Universidade Federal do Paraná – Graduação em Engenharia Cartográfica e de Agrimensura – Av. Cel. Francisco H. dos Santos, 100, 81530-000 Curitiba, Paraná ([luiz.carneiro@ufpr.br](mailto:luiz.carneiro@ufpr.br))

<sup>2</sup> Universidade Federal do Paraná – Programa de Pós-graduação em Ciências Geodésicas – Av. Cel. Francisco H. dos Santos, 100, 81530-000 Curitiba, Paraná ([luciene@ufpr.br](mailto:luciene@ufpr.br); )

<sup>3</sup> Universidade Federal do Paraná – Centro de Pesquisas Aplicadas em Geoinformação – CEPAG – Av. Cel. Francisco H. dos Santos, 100, 81530-000 Curitiba, Paraná ([leonardo.ercolin@ufpr.br](mailto:leonardo.ercolin@ufpr.br))

## ABSTRACT

The UFPR CampusMap Project has as its main objective map the UFPR Campi, not only the outdoor environment but the *indoor* as well. The *indoor* maps are represented in two ways: blueprints and schematics. The second one is obtained manually based on the blueprint. Therefore, the main objective of this research project is to elaborate a QGIS plugin in which the hallway skeleton is extracted in an automated way, using the “Voronoi Polygons” tool, and the next goal is to create a Schematic Map; This result is necessary to create a route in an *indoor* environment.

**Keywords:** Schematic, Indoor, Plugin

## INTRODUÇÃO

O crescimento da utilização de dados geoespaciais em dispositivos móveis, tem resultado no surgimento de aplicações voltadas para o mapeamento *indoor* (POTIGIETER, 2015). As aplicações, em geral, têm como principal finalidade auxiliar os usuários a se localizarem em ambientes *indoor* pouco conhecidos, pelo fato de se sentirem facilmente perdidos nos mesmos (SI, ARIKAWA, 2015).

O UFPR CampusMap ([www.campusmap.ufpr.br](http://www.campusmap.ufpr.br)) é um projeto de pesquisa e extensão que tem por objetivo mapear os diversos Campi da Universidade Federal do Paraná (UFPR). Os resultados do mapeamento são apresentados em uma plataforma web, juntamente com outros dados, como por exemplo, ortoimagens e dados da rede topográfica da UFPR. Uma das suas funcionalidades consiste em traçar rotas entre diferentes pontos dos campi. Para a geração das rotas é utilizado uma estrutura de rede, composta de linhas e nós, neste trabalho denominada mapa esquemático. Um mapa esquemático é uma simplificação de uma planta baixa, composto de linhas e nós, que correspondem respectivamente aos corredores e salas. Atualmente, o processo de construção de um mapa esquemático é manual e por isso, leva tempo para ser concluído. Assim, esta pesquisa tem por objetivo propor um método para automatizar o processo de construção de mapas esquemáticos por meio da criação de um plugin para o QGIS. Esse plugin possibilita extrair linhas a partir dos corredores e nós a partir da posição das portas através do modelador gráfico do QGIS. Os resultados obtidos mostraram que é possível gerar um mapa esquemático através

desse modelo, mas é necessária uma edição manual a fim de melhorar o aspecto da representação gerada.

## METODOLOGIA

O trabalho foi realizado no Laboratório de Cartografia e SIG da Universidade Federal do Paraná (UFPR), e o software escolhido para tal foi o QGIS. O QGIS é um software *Open Source*, ou seja, é gratuito e permite alterações feitas pela comunidade em seu código fonte. O conjunto de dados utilizados nesse trabalho foi cedido pelo Centro de Pesquisas Aplicadas em Geoinformação (CEPAG).

## PREPARAÇÃO DOS DADOS

As plantas baixas estão armazenadas em arquivos em formato shapefile, contendo as salas, corredores e o limite das edificações. Para dar início ao processo de esquematização da planta baixa, a primeira etapa consistiu em extrair da planta baixa apenas as feições do tipo “corredor” para uma camada vetorial própria para facilitar a manipulação dos dados. Para isso, aplicou-se um filtro, via tabela de atributos, onde todas as feições foram selecionadas para uma nova camada, denominada “vetorCorredor”. A etapa seguinte consistiu em utilizar o modelador gráfico do QGIS que apresenta, além da camada vetorial dos corredores, outra camada vetorial de pontos, que representa as portas, chamada de “pontoPortas”. O modelo completo encontra-se na Figura 1.

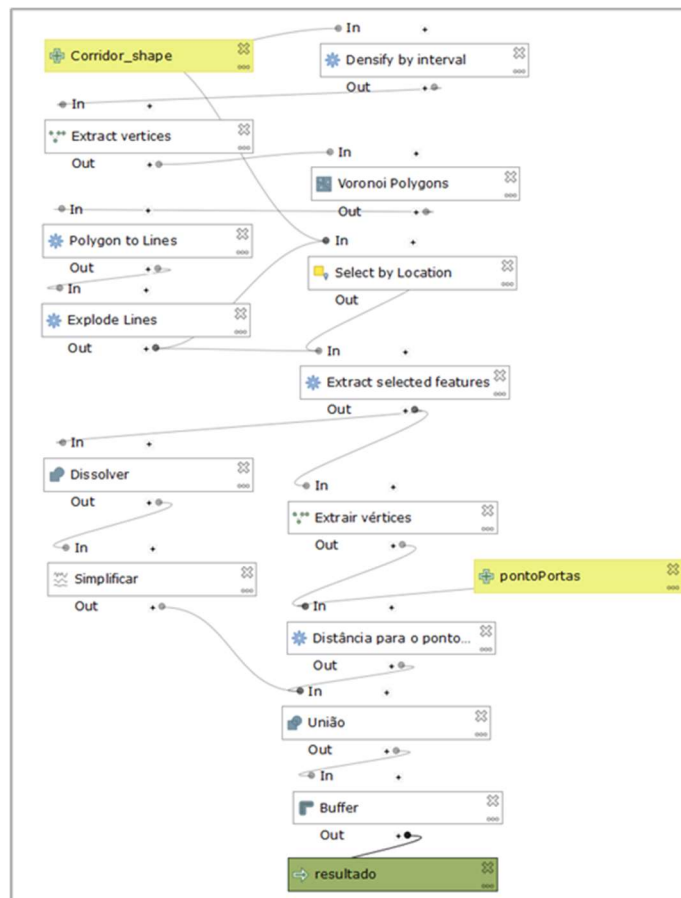


Figura 1 – Modelo proposto

Depois de finalizada a etapa de seleção de feições, a etapa seguinte consistiu em adicionar vértices às arestas dos polígonos de acordo com um intervalo definido (função “*Densify by interval*”). Isto é feito para aumentar a precisão na geração e suavização dos polígonos de Voronoi, melhorando a apresentação do mapa esquemático. Como se trata de corredores, o intervalo definido foi de 0,1m (10 centímetros).

Após criados os vértices, estes são extraídos para uma nova camada vetorial temporária, e com esses vértices são gerados os polígonos de Voronoi (Figura 2). Um diagrama de Voronoi é um conjunto de regiões em torno de pontos no plano euclidiano. Estas regiões são as que estão mais próximas de um dos pontos do que de qualquer outro ponto. Nesse caso, os pontos para formar o diagrama são os vértices extraídos da densificação por intervalo das linhas dos corredores. A sua construção depende da distância entre os pontos, formando delimitações no centro dela. Após a geração do diagrama, nota-se uma grande quantidade de linhas que são desnecessárias para realizar o sistema de rotas, portanto é necessária uma filtragem de elementos que não serão utilizados. Para isso, são selecionadas apenas as linhas que estão inteiramente dentro da geometria do corredor (função “*Select by location*”), e as outras por sua vez são eliminadas a fim de se criar o mapa esquemático em si, mostrado na Figura 3.

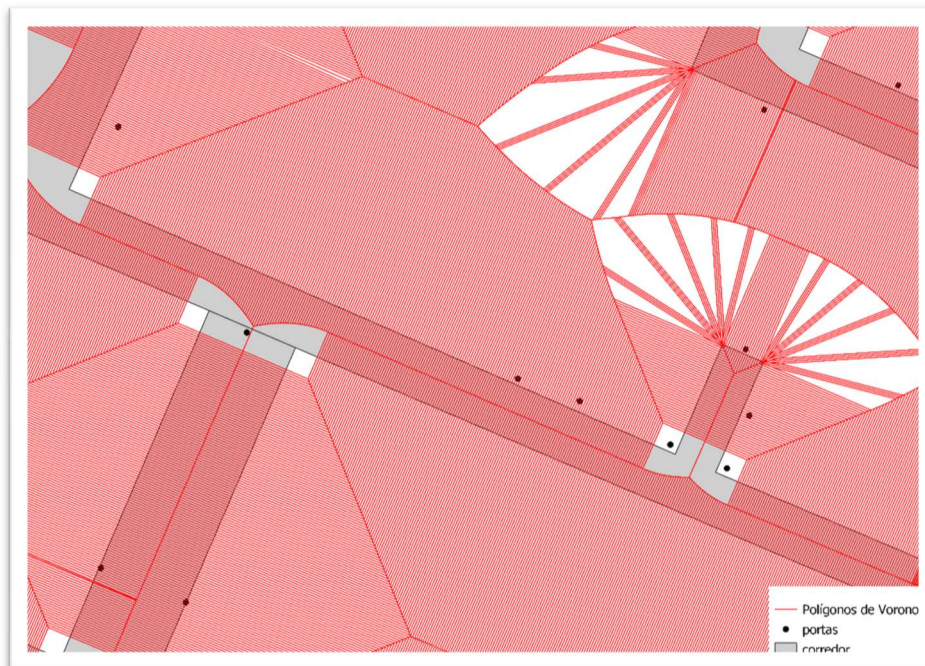


Figura 2 – Polígonos de Voronoi

Com o resultado obtido na etapa anterior, é possível visualizar as linhas do mapa esquemático. Entretanto, é necessária a exclusão das linhas auxiliares que não serão utilizadas. Para isso, é feita a transformação dos polígonos em linhas, e essas linhas por sua vez são separadas para serem selecionadas individualmente (função “*Explode Linhas*”). A partir disso, é possível selecionar e extrair apenas as linhas que estão inteiramente no polígono dos corredores, utilizando as funções “*Selecionar por localização*” e “*extrair feições selecionadas*”, respectivamente. Em seguida, as linhas

são dissolvidas, ou seja, o que estava separado para uma melhor seleção, agora é unido em um elemento, no qual é aplicada a função “simplificar”, a fim de obter um mapa esquemático mais agradável à visão do usuário.

Para unir o corredor às portas, utiliza-se a função “distância para o ponto central mais próximo (“linha para ponto central”)", a qual une o ponto das portas para o vértice mais próximo do esqueleto do polígono. Em sequência, é feita a união dessa camada resultante, com a camada do corredor em si. Agora com apenas uma camada, aplica-se um buffer de 0,1m com função estética, e o programa se encerra, apresentando o seguinte resultado, apresentado na Figura 3:

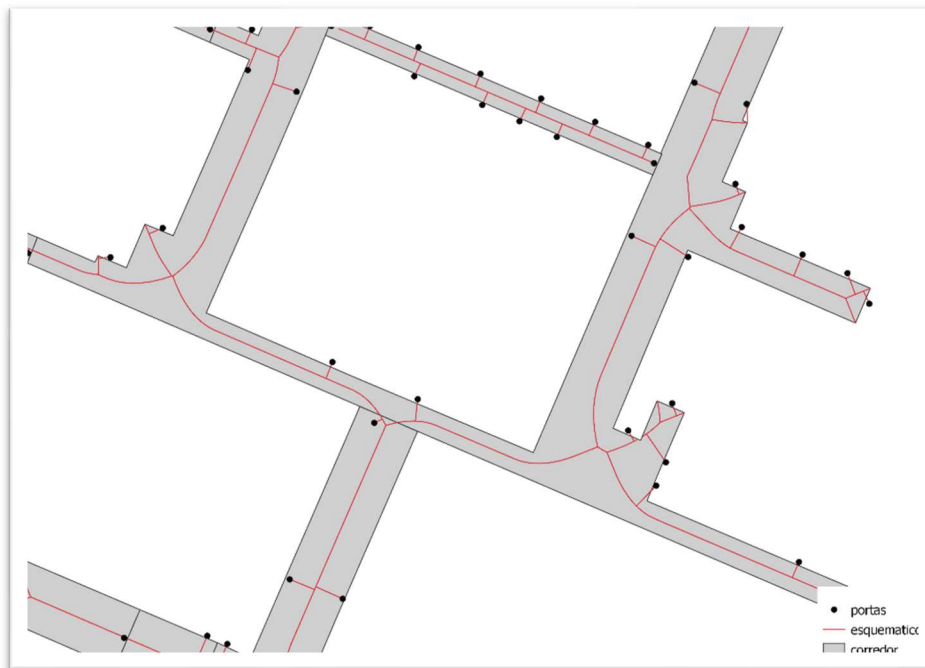


Figura 3 – Mapa Esquemático

### CONSTRUÇÃO DO PLUG-IN

Para a construção do plug-in a partir de um modelo feito no QGIS, utilizou-se a ferramenta chamada *Plugin Builder*, que está disponível na aba de complementos do software. Com esta ferramenta, é possível construir um complemento que pode ser publicado no repositório oficial de plug-ins do QGIS. O modelo gerado no formato .model3 foi exportado para um arquivo de formato .py (Python), que é a linguagem de complementos do QGIS.

Primeiramente, é necessário inserir as informações básicas do complemento, como nome, descrição e nome do autor. Como mostrado nas figuras 4 e 5, na opção “*Template*”, seleciona-se o item *Processing Provider*, visto que o plug-in será criado com base em um modelo feito no modelador gráfico. Também é necessária uma página da web e um *bugtracker* para o complemento; nesse caso utilizou-se para ambos o GitHub, pelo fato de ele já apresentar uma aba de *bugtracking* própria (Issues), disponível em [https://github.com/LuizOt/indoor\\_schematic](https://github.com/LuizOt/indoor_schematic). A figura 4 mostra os detalhes da criação do plugin.

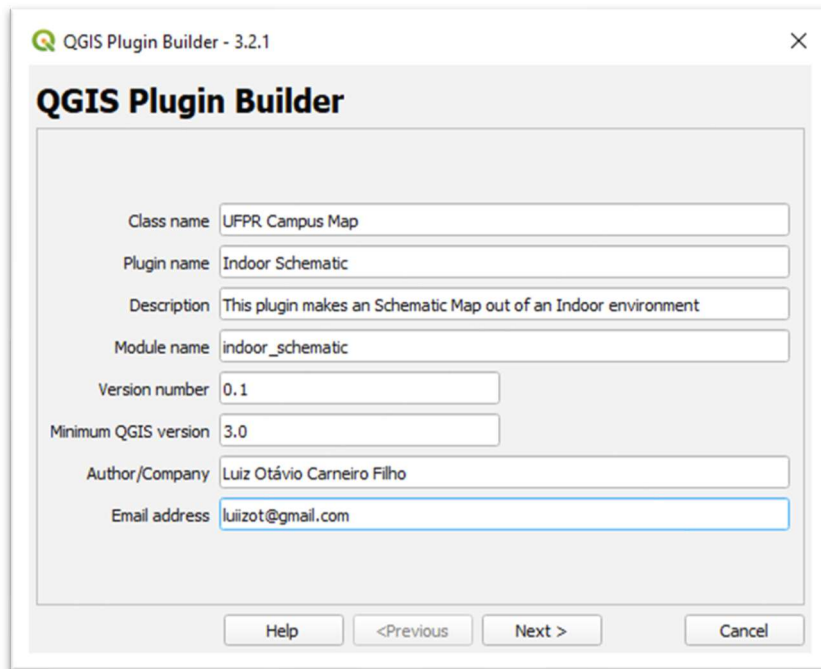


Figura 4 – *Plugin Builder*

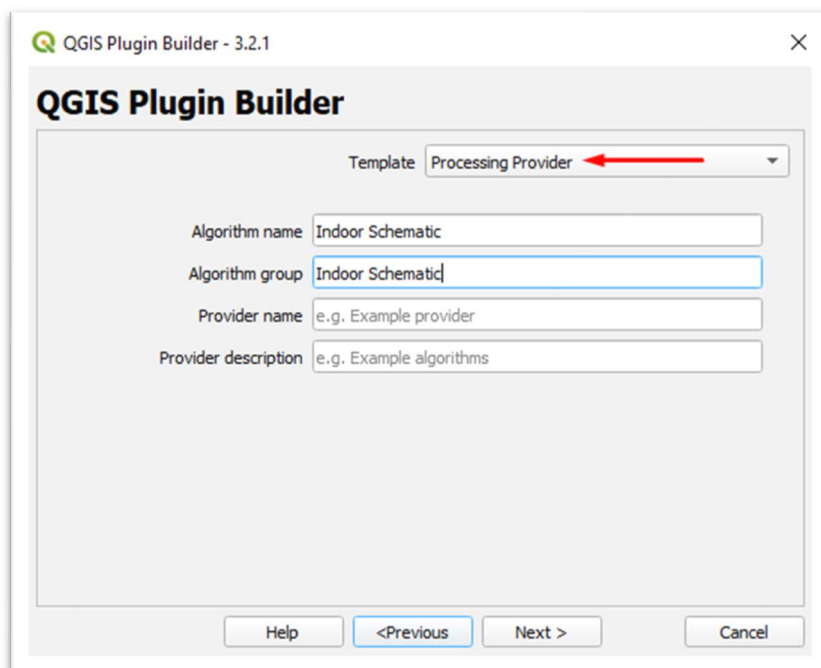


Figura 5 – *Processing Provider*

Em seguida, são necessárias algumas alterações na estrutura criada para que o plug-in funcione com base no modelador gráfico, as quais são citadas a seguir:

- Substituir o arquivo com o algoritmo do plugin pelo arquivo python do modelo feito no QGIS (model);
- No arquivo do provedor do plugin, na linha 34, é necessário informar ao programa que o arquivo que será importado é o arquivo do algoritmo substituído anteriormente e a classe que está contida no mesmo.

## RESULTADOS E DISCUSSÃO

Após obtido o resultado exibido na Figura 3, nota-se que ainda existem alguns problemas na geração do mapa esquemático que ainda estão sendo pesquisados, como a formação de um “v” no final de cada corredor. Em geometrias mais complexas, as linhas formam curvas enquanto deveriam ser retas; para resolver esse problema, uma opção é dividir o polígono dos corredores em diversos retângulos, devido ao fato de que em geometrias mais simples, as linhas resultantes ficam coerentes com o resultado esperado.

Analisando o produto gerado pelo complemento, em comparação com o mapa esquemático gerado manualmente, nota-se que em certas geometrias os resultados são discrepantes. Isso pode acontecer devido à vários fatores, entre eles a presença de geometrias inválidas na camada de entrada do processamento. Após a execução do plugin e análise dos resultados, realizou-se uma verificação de topologia, e onde houve erros topológicos, foi utilizada a opção do software para considerar essas geometrias como válidas, o que resultou em uma diminuição dos problemas.

O plug-in publicado encontra-se disponível no repositório oficial de plugins do QGIS, no seguinte endereço: [https://plugins.qgis.org/plugins/indoor\\_schematic/](https://plugins.qgis.org/plugins/indoor_schematic/) .

## CONCLUSÃO

Ao final do trabalho, nota-se um grande progresso no processo de automatização da esquematização de uma planta baixa. Entretanto, ainda é preciso resolver alguns problemas a fim de obter um resultado final mais próximo do processo de geração manual. Dentre as melhorias previstas para o modelo estão a edição do código da função dos polígonos de Voronoi utilizando Python, para que as linhas que se uniram com os cantos dos corredores sejam eliminadas, e o desenvolvimento de um algoritmo para dividir os corredores em retângulos de forma automatizada, para analisar o comportamento das linhas.

## AGRADECIMENTOS

Os autores agradecem ao CNPq pela concessão da bolsa de Iniciação Científica e pela bolsa Produtividade em Pesquisa (processo 310312/2017-5). Agradecem também ao CEPAG – Centro de Pesquisas Aplicadas em Geoinformação ([www.cepag.ufpr.br](http://www.cepag.ufpr.br)) pela cessão dos dados utilizados no projeto.

## REFERÊNCIAS

SI, Ruochen., & ARIKAWA, Masatoshi. A Framework of Cognitive Indoor Navigation Based on Characteristics of Indoor Spatial Environment. ISPRS – International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2015.

POTIGIETER, Pierre. Indoor mapping: beyond the front door. EE Publishers (<http://www.ee.co.za>), p 1–9, 2015.