# ARTIGO TÉCNICO/CIENTÍFICO

**DESENVOLVIMENTO DE COMPLEMENTO EM SISTEMA DE INFORMAÇÃO GEOGRÁFICA PARA REALIZAÇÃO DE PROJETOS GEOMÉTRICOS DE ESTRADAS**

(1)Matheus Fillipe Araújo Galvão

(2)Reynaldo Furtado Faria Filho

(1)Estudante de Engenharia Civil - Universidade Federal de Viçosa, Rio Paranaíba

(2)Professor Adjunto do curso de Engenharia Civil - Universidade Federal de Viçosa, Rio Paranaíba

Presidente da Banca: Prof. Reynaldo Furtado Faria Filho

Membro 1: Profa. Adriana Zanella Martinhago

Membro 2: Prof. Vinícius Gonçalves de Almeida

**16 de Dezembro de 2019**

**RESUMO:** Os transportes são um fator fundamental para a economia de qualquer país e no caso do Brasil o modal rodoviário é o mais utilizado e recebe mais investimentos. Em 2005 o Governo Federal estabeleceu prioridade para utilização do software livre com objetivo de aliar eficiência com custo-benefício. Nesse contexto se encaixa o QGIS como software de informações geográficas com código aberto e com a possibilidade de ser estendido por meio de complementos escritos na linguagem python, que é de alto nível e suporta múltiplos paradigmas de programação. Nesse artigo é apresentado o desenvolvimento de um complemento para o QGIS escrito em python para auxiliar em todos os passos básicos de um projeto geométrico de estradas, deste o traçado horizontal, curvas, até o calculo de volumes de corte e aterro. A intenção foi desenvolver uma alternativa de código aberto e totalmente gratuita semelhante a softwares já conceituados no mercado como o Civil 3D da AutoDesk e o topoGRAPH de Bentley.

**PALAVRAS-CHAVE:** concordância horizontal, concordância vertical,complemento, python, QGIS.

**GEOGRAPHIC INFORMATION SYSTEM PLUGIN DEVELOPMENT FOR THE GEOMETRIC DESIGN OF ROADS**

**ABSTRACT:** Transport is a fundamental factor for the economy of any country and in Brazil road transportation is the most used mode and the one that receives more investments. In 2005 the Federal Government established the priority on using free software for cost efficiency. QGIS comes to this context as an open source geographic information system software which supports being extended by plugins written in the high-level programming language python. This paper presents the development of a QGIS plugin written in python with the goal of assisting in all the basic steps of geometric road design, beginning with the horizontal layout definition, through curves, up to the volumes of cut and embankment, with the ambition of developing a completely free and open source alternative for similar and already popular software in the industry such as AutoDesk Civil 3D and Bentley topoGRAPH.

**KEYWORDS:** design, geometric, road, sizing, plugin, python, QGIS.

# **IN****TRODUÇÃO**

O Brasil para ser competitivo frente aos demais países, nas diversas áreas da economia, deve ter sua infraestrutura de transportes como um fator estratégico fundamental (VASCONCELOS & BASSO, 2008). De acordo com o boletim estatístico da Confederação Nacional do Transporte (CNT, 2018) o modal rodoviário possui atualmente, no Brasil, a maior representatividade entre os modais existentes, com finalidade no transporte de cargas e passageiros (Tabela 1).

Tabela 1: Porcentagem de representatividade entre os modais existentes no Brasil.

|  |  |
| --- | --- |
| Modal | Porcentagem |
| Rodoviário | 61,1 |
| Ferroviário | 20,7 |
| Aquaviário | 13,6 |
| Dutoviário | 4,2 |
| Aéreo | 0,4 |

Fonte: CNT (2018).

O transporte rodoviário, por ser o mais utilizado, recebe investimentos constantes para sua manutenção, ampliação, bem como criação de novas rodovias. Para a criação de novas rodovias deve-se realizar diversos estudos/projetos: estudos de planejamento de transporte; estudos de tráfego; estudos geológicos; estudos hidrológicos; estudos topográficos; projeto geométrico; projeto de obras de terra; projeto de terraplenagem; projeto de pavimentação; projeto de drenagem; projeto de obras de arte; projeto de viabilidade econômica; projeto de desapropriação; projetos de interseções, retornos e acesso; projeto de sinalização; projeto de elementos de segurança, dentre outros (PONTES FILHO, 1998).

O projeto geométrico de uma estrada é composto pelos elementos físicos, como comprimento de retas, largura de pista, raios de curvas e rampas, que estão diretamente associadas às características de operação dos veículos (frenagem, velocidade, distância de visibilidade, dentre outros). Assim, na maioria dos casos o desenvolvimento do projeto geométrico é uma sequência estruturada por rotinas que visam definir os parâmetros geométricos, bem como calcular a geometria. A repetição destas rotinas fornece várias soluções para análise e, a escolha de uma considerada melhor possibilita o desenho do projeto (OLIVEIRA JUNIOR, 2007).

No mercado existem diversos softwares disponíveis para a geração de projeto geométrico de estradas, entretanto o custo de aquisição, principalmente para o ensino nas Universidades, é elevado. Pode-se citar alguns destes softwares disponíveis no mercado que possibilitam a realização de projeto geométrico de estrada: TopoCal, Civil 3D, topoGRAPH, DataGeosis, TCP-MDT, InRoads, Posição, dentre outros (MENEZES et al., 2014).

Em 2005 o Governo Federal estabeleceu prioridade para utilização do software livre com objetivo de aliar eficiência com custo-benefício (GUIA LIVRE, 2005). Assim, além de um software livre poder ser aplicado na geração de um projeto geométrico de uma rodovia que será implantada, este poderá ainda ser de grande utilidade para o ensino de qualidade nas Universidades com um custo baixo para às mesmas.

Neste contexto está inserido o QGIS, anteriormente chamado de Quantum GIS, que é um software de Sistemas de Informações Geográficas. O QGIS é um Sistema de Informação Geográfica (SIG) de Código Aberto licenciado segundo a Licença Pública Geral GNU. O QGIS é um projeto oficial da Open Source Geospatial Foundation (OSGeo). Funciona em Linux, Unix, Mac OSX, Windows e Android e suporta inúmeros formatos de vetores, rasters e bases de dados e funcionalidades.” (OSGEO et al., 20--).

Assim, o objetivo deste trabalho foi o de aliar as capacidades de geoprocessamento do QGIS com uma base de código que cobre cada uma das etapas de um projeto geométrico de estradas, criando um complemento com fins educacionais a priori.

**2. REFERENCIAL TEÓRICO**

1. **Projeto geométrico de estradas**

Antes de projetar os parâmetros de uma rodovia deve-se esgotar todas as possibilidades de desenvolvimento de um traçado. Para isso, o Departamento Nacional de Infra-estrutura de Transportes (DNIT) disponibilizou a Instrução de Serviço IS-207 que trata dos Estudos Preliminares de Engenharia para Rodovias, ou seja, estudos de traçado (DNIT, 2006). Para isso, deve-se elaborar os estudos que abrangem as seguintes tarefas:

a) Definir as diretrizes tecnicamente possíveis da rodovia;

b) Determinar a viabilidade física das alternativas indicadas como sendo as adequadas pelos estudos de tráfego para a ligação rodoviária proposta;

c) Definir certas soluções básicas para a elaboração desembaraçada dos trabalhos mais detalhados nos estudos posteriores;

d) Estimar os custos aproximados de construção e de desapropriação para fins de avaliação econômica e financeira.

Após a fase de estudo dos traçados faz-se a elaboração do projeto geométrico. De acordo com a Instrução de Serviço IS-208, do DNIT o projeto geométrico deve ser elaborado em duas fases (DNIT, 2006):

a) FASE DE PROJETO BÁSICO: deve ser elaborado a partir dos estudos topográficos realizados segundo o que dispõe a Instrução de Serviço IS-204. Já as características geométricas mínimas do projeto geométrico, serão norteadas pelos Estudos de Tráfego, e deverão atender as recomendações do Manual de Projeto Geométrico de Rodovias Rurais DNER - 1999. O projeto geométrico deve ser elaborado detalhadamente de maneira a apresentar condição de ser locado na fase de projeto. No projeto geométrico devem constar: a) Quadro de características técnicas e operacionais; b) Projeto em planta, na escala de 1:2000 (composição das curvas horizontais; elementos cadastrais; interseções; pontes e viadutos e bueiros); c) Projeto em perfil, nas escalas de 1:2.000 (H) e 1:200 (V) (composição das curvas verticais; rampas; pontes e viadutos e bueiros) e d) Seções transversais típicas da plataforma.

b) FASE DE PROJETO EXECUTIVO: deve ser elaborado a partir dos estudos topográficos realizados segundo o que dispõe a Instrução de Serviço IS-205. Este projeto deverá constituir-se de: a) Projeto planialtimétrico, nas escalas de 1:2.000 (H) e 1:200 (V); b) Determinação das seções transversais do projeto, nas escalas de 1:200 ou 1:100 e c) Detalhamento dos elementos especiais do projeto como: retornos e acessos em nível; terceiras faixas de tráfego; tapers.

As rodovias por ser um ente tridimensional podem ter seus elementos geométricos projetados nas três dimensões. A linha (eixo) que representa a via é projetada em um plano horizontal e representada em uma planta, enquanto que o projeto em perfil tem seu dimensionamento feito a partir de um plano vertical, possibilitando a definição do greide que é a representação do eixo da pista no plano vertical (OLIVEIRA JUNIOR, 2007). Os principais elementos geométricos de uma estrada são apresentados na Figura 1 - Anexo A (PONTES FILHO, 1998).

1. **Projeto geométrico de estradas desenvolvido com auxílio do computador**

Menezes et al. (2014) fizeram uma análise comparativa dos softwares topoGRAPH 98 SE e TCP-MDT (versão 6.7) que são utilizados para o desenvolvimento de projetos geométrico de estradas. O primeiro (topoGRAPH) possui mais de duas décadas de existência e é bastante difundido no Brasil, enquanto que o segundo (TCP-MDT) é de origem Europeia. Ao final do trabalho os autores concluíram que “*embora o software TCP-MDT, se encontra em adaptação às normas brasileiras, demonstrou-se de forma geral, ser um programa simples com recursos robustos e práticos*”, enquanto que o “*topoGRAPH não apresentou suas ferramentas de projeto tão explícitas quanto ao TCP-MDT, além de deixar a desejar na sua visualização em 3D*”. Os autores salientam que o topoGRAPH está em conformidade com as normas brasileiras.

Campos Neto et al. (2012) compararam os softwares AutoCAD Civil 3D e o Microstation V8i com o módulo InRoads para a geração do projeto geométrico de uma estrada. Os autores fizeram ainda o projeto de forma manual. Ao final do trabalho ficou evidente que o processo computacional é superior ao manual em relação ao tempo gasto para desenvolver um projeto e em relação à qualidade do produto final. Quanto aos softwares os autores concluíram que o Bentley InRoads se mostrou superior ao AutoCAD Civil 3D no que se diz respeito ao desempenho.

No que se refere a software Livre, para realização de projeto geométrico de estradas, o autor Silva Júnior (2003) desenvolveu um software cuja metodologia divide a elaboração do projeto geométrico automatizado em cinco etapas básicas:

Etapa 1 - Definição Topográfica para o Projeto;

Etapa 2 - Aquisição e processamento dos Dados Topográficos;

Etapa 3 - Criação do Modelo Digital do Terreno;

Etapa 4 - Criação do Modelo Digital do Projeto; e

Etapa 5 - Emissão de relatórios, desenhos e medições.

O autor Oliveira Junior (2007), implementou outras funções ao software desenvolvido por Silva Júnior (2003). Como o trabalho de Silva Júnior (2003) centrou-se na aquisição de dados, Oliveira Junior (2007), procurou desenvolver seu sistema a partir desse ponto, ou seja, fez a incorporação dos dados já levantados. Para isso, Oliveira Junior (2007) utilizou a linguagem de programação JAVA por ser uma linguagem portátil, flexível e possível de ser utilizada em diferentes sistemas operacionais. Além da linguagem JAVA o referido autor utilizou o sistema JUMP por ser um pacote com solução para alguns problemas, como a representação gráfica do projeto, além de possibilitar a realização do projeto inteiramente georreferenciado, dentro de um sistema de informações geográficas, aumentando consideravelmente sua potencialidade.

Oliveira Junior (2007) relata que o seu software teve o intuito de:

[…] desenvolver um sistema novo, mas que não se esgotasse, ou seja, que não obrigasse a um próximo pesquisador iniciar tudo novamente. Para tanto, ele teve como características principais ser um sistema flexível, de fácil reutilização, possuir um modelo teórico, um projeto de software independente da linguagem de implementação e ser documentado para possibilitar o entendimento do que foi proposto. (JUNIOR, 2007)

1. **Ambiente** **de programação**
   * 1. **Python**

Python é uma linguagem de programação criada por Guido van Rossum em 1991. A linguagem suporta múltiplos paradigmas de programação, como a programação estruturada funcional e orientada a objetos. Criada sobre a licença de código livre e multiplataforma, o Python está presente nos principais sistemas operacionais como Windows, Linux e OSX (PYSCIENCE, 2016).

A linguagem Python é amplamente utilizada em projetos científicos devido a facilidade de desenvolvimento e integração com sistemas legados e bibliotecas providas por outras linguagens. Além disto, é possível ter uma boa performance sendo muitas vezes comparáveis a C++, pelo fato da plataforma ser simples fazendo com que consiga fazer otimizações que seriam mais trabalhosas em outras linguagens (PYSCIENCE, 2016).

* + 1. **Qt**

Utilizado por diversas empresas e projetos de código aberto, o Qt é considerado um dos mais versáteis frameworks gráficos. Desenvolvido em 1995 pela empresa norueguesa Trolltech, posteriormente pela equipe da Nokia e atualmente pela Digia. Tem como principal atrativo a premissa de código universal, no qual o programador não precisara se preocupar como funciona a gestão de elementos em interfaces de cada sistema operacional. Pois o mesmo é feito pela plataforma, necessitando apenas ter um adaptador para cada sistema, para gerar os executáveis para cada um deles. Como se trata de um projeto, que embora tenha fins comerciais, possuir código livre para melhorias de integração e funcionamento faz com que ele seja largamente utilizado em soluções de código aberto e em diversas mídias (NOKIA, 2009).

Desenvolvido em c++ com suporte a diversas linguagens como Python, C#, Java, R, Ruby entre outros. Além do suporte para dispositivos embarcados e móveis, faz do framework uma das soluções mais rápidas e simples para desenvolvimento de software multi-plataforma. Sua arquitetura se baseia na criação de interfaces que utilizam o QtDesigner, na qual gera-se um código específico para a linguagem escolhida. A linguagem a C++ é gerada nativamente, e com isso é possível fazer todo monitoramento de eventos e respostas da interface, sem programar a mesma de maneira direta. Em seguida pode-se gerar executáveis para cada arquitetura de hardware e de sistema operacional (HÜBSCHLE et al., 2011).

Essa biblioteca também é utilizada como a base de toda a interface gráfica para o QGIS e consequentemente a mais recomendada e viável para o desenvolvimento das interfaces para complementos dentro desse software, já que já faz parte de seu ambiente padrão.

* + 1. **Pyqtgraph**

*Pyqtgraph* é uma biblioteca para representação gráfica e criação de interfaces desenvolvida em *python* usando PyQt4 e *numpy*. Seu objetivo é a utilização em aplicações matemáticas, científicas ou na engenharia. Apesar de ter sido completamente escrita em python, seu desempenho é bom graças a ajuda do *numpy* para o processamento de números e a *Framewor*k *GraphicsView* do Qt para visualização acelerada. Essa biblioteca é distribuída na licença *open source* MIT (PYQTGRAPH, 2018).

Com o auxílio dessa biblioteca podem ser gerados gráficos e interfaces de desenho em 2D que o QGIS, por si só não disponibiliza tais como soluções para a realização de traçados verticais e para a elaboração das sessões tipo e visualizações de perfis.

* + 1. **QGIS**

O QGIS, antes conhecido como Quantum GIS, é um software gratuito, *open source* e de multilataforma, sendo compatível com os sistemas operacionais Windows, Linux, Mac e Android (em estado de desenvolvimento) sendo dedicado à visualização, edição e análise de dados geoespaciais. É um projeto oficial da “*Open Source Geospacial Foundation*” (OSGeo) licenciado sobre a GNU suportando várias modalidades de computação vetorial com o auxílio de *rasters* e outros formatos de banco de dados. Foi criado por Gary Sherman em 2002 e escrito e C++ fazendo uso extensivo da biblioteca Qt e mantendo até hoje um programa de manutenção e lançamento composto majoritariamente por programadores voluntariados. A partir do lançamento da versão 2, o QGIS obteve suporte ao desenvolvimento de plugins com a linguagem python de programação contando com um repositório oficial online com os mais diversos tipos de ferramentas que podem ser facilmente instaladas (QGIS, 2018).

O QGIS não possui atualmente aplicações especificas para o desenvolvimento de um projeto geométrico de estradas. Entretanto, possui diversas ferramentas implementadas que podem permitir o seu desenvolvimento. O projeto QGIS iniciou com o intuito de ser um visualizador gratuito de bases de dados geográficas que pudesse ser instalado em sistemas operacionais livres. O projeto foi iniciado em 2002 (MANGHI et al., 2011). Atualmente o software pode ser instalado em diversas plataformas como Unix, GNU/Linux, Mac OsX e MS Windows.

**3****. METODOLOGIA**

Para implementação do novo sistema foi necessário o estudo dos *plugins* existentes para QGIS e também o domínio na criação desse tipo de software. Por padrão os *plugins* para essa plataforma são desenvolvidos na linguagem *python* com interface gráfica Qt. O uso do QGIS como base desse projeto, ao invés de uma solução autônoma, deve-se à quantidade de ferramentas de geoprocessamento que já se encontram na API do QGIS como por exemplo:cálculos geométricos diversos no plano, leitura de pixels em *rasters,* que são de fácil acesso para o desenvolvedor.

O projeto foi escrito na versão 3.6 da linguagem python para ser compatível com a terceira versão do QGIS. A biblioteca de interface gráfica (GUI) utilizada é a PyQt5 que está presente dentro do próprio ambiente python padrão do QGIS, o que facilita o desenvolvimento e reduz o tamanho final do *plugin*. Essa biblioteca contém os *bindings* do Qt para python, tornando possível a criação de interfaces gráficas em *plugins*.

Inicialmente foram identificados os requisitos necessários para o desenvolvimento do projeto geométrico de uma estrada e em seguida foi feita uma modelagem gráfica da solução para facilitar o trabalho na etapa seguinte que é a implementação da mesma. Esta implementação foi basicamente a transformação da modelagem para código “python” integrando com a interface gráfica “Qt”.

Foi desenvolvida a leitura dos dados de altimetria ou cotas. Nela é utilizado um modelo digital no formato “Geotiff” que é um *raster* onde os valores dos pixels indicam os valores das elevações. Os valores extraídos desses arquivos podem, opcionalmente, ser interpolados durante a leitura de acordo com a intenção do usuário. Esta interpolação pode acarretar em maiores gastos computacionais.

Por fim foi realizado o teste das funcionalidades, por meio de estudos de casos com testes desenvolvidos em conjunto com o profissional da área. A partir dos relatos dos usuários foram realizadas as alterações para melhorar o sistema desenvolvido. O complemento está baseado na possibilidade de uso em múltiplos sistemas operacionais, visto que foi testado no Windows 10, Windows 7, Linux Ubuntu 16.04 e 18.04, tendo somente o QGIS 3 como dependência.

**3.1 Gestão e armazenamento dos dados de projeto**

A modelagem no plano horizontal baseou-se principalmente nas configurações básicas e na criação de estacas e curvas. Estas pertencem ao mesmo controle, por se tratar de recursos semelhantes em nível de controle, no entanto, se separam em outras camadas, pois as ações e a interação ocorre de maneira diferente. A modelagem do perfil vertical se deu a partir da tabela greide e o perfil transversal a partir do relevo da sessão, sessão tipo e perfil transversal.

Para o armazenamento dos dados foi definida a necessidade de ser um sistema portátil que possa ser facilmente manipulado sem gastar muitos recursos computacionais. Seguindo esses requisitos a melhor solução encontrada foi o SQLite, por atender as necessidades, possuir fácil integração com a plataforma QGIS e por ser leve, mesmo sendo um sistema gerenciador de banco de dados bastante robusto. Como apresentado na Figura 2 (Anexo A), a estratégia utilizada para manipulação de dados foi separá-los em entidades relacionadas representadas pelos quadros dessa imagem, sendo elas: estacas, curvas horizontais, sessão tipo, greide, relevo, curva vertical. Os dados das configurações principais são armazenados na tabela “PROJECT”, tais como a classe de projeto e as limitações de inclinação e velocidade.

A fim de facilitar o carregamento e edição do perfil horizontal e suas curvas, adotou-se em redundância o formato de arquivo *geopackage*, que também contém um banco de dados SQLite podendo ser instanciado dentro do QGIS no formato de uma *layer*. Com o mesmo objetivo de otimização para as sessões transversais desenvolveu-se uma biblioteca para armazenar os dados geométricos prismoides que fosse capaz de ser serializada pela biblioteca padrão do python “pickle”. Com isso pode-se gerar um arquivo único que possa ser salvo no disco e instanciado na memória rapidamente contendo todas as informações massivas da sessão transversal. Esse arquivo recebeu a extensão “.prism”.

A fim de simplicidade e praticidade, todos esses arquivos ficam armazenados em uma pasta compactada “.zip” que pode ser compartilhada e importada em outra máquina com o QGIS e com o complemento. Esse arquivo compactado também fica vinculado e com seu caminho salvo no arquivo de projeto próprio do QGIS. A árvore de arquivos de projeto é representada pela Figura 3 (Anexo A). Vale observar que o nome dos arquivos é dado em função do id do traçado correspondente que é armazenado pela “TABLEESTACA” (Figura 2 - Anexo A), já que o mesmo projeto pode possuir múltiplos traçados. O complemento faz uso das pastas temporárias do sistema operacional para descompactar esses arquivos durante suas diversas operações.

**3****.2 Traçado e curvas horizontais**

A definição do traçado horizontal pode ser feita criando uma ou importando uma *layer* no QGIS. Uma *layer* pode ser gerada por meio de arquivos do tipo *shapefile*, *dxf* (CAD), *geopackage,* enfim, qualquer arquivo que seja suportado pelo QGIS será suportado pelo complemento desde que seja do tipo linha vetorial, mesmo possuindo múltiplos atributos. O QGIS se encarrega de converter todos esses formatos geométricos para um tipo único “QgsVectorLayer” de onde podem ser extraídas todas as informações geométricas necessárias.

A Tabela 2 representa os tipos de dados que estão contidos na *layer* de desenho. A coluna atributo mostra o identificador para cada parte da geometria que pode possuir campos diferentes, o que é comum dos tipos de arquivos com informação geográfica. O complemento pode importar *layers* com esses campos já preenchidos ou gerar a partir de outra *layer.* O campo tipo representa o tipo de geometria. Qualquer valor de texto que inicia com “T” será interpretado como tangente, “C” para curva circular simples e “E” para espiral.

O algoritmo de importar *layers* segmenta o desenho do traçado horizontal entre o intervalo de estacas definido pelo usuário nas configurações, que é por padrão 20 metros. Em cada ponto desses fica definido uma estaca sem descrição. Em cada ponto inicial, final ou de interseção entre duas tangentes define-se uma estaca intermediária “PI” (ponto de interseção). Em pontos notáveis tais como nas mudanças de tipo de atributo do desenho, o que pode significar o início da parte de transição de uma curva ou da parte circular, define-se uma estaca intermediária que recebe a descrição “TS”, “SC”, “CS” ou “ST”, onde “T” significa tangente, “S” para espiral e “C” para trecho circular. O campo descrição é opcional com a intenção de permitir o usuário fazer anotações para cada trecho.

Para cada uma das estacas obtidas calcula-se as coordenadas de acordo com a fonte CRS configurada no QGIS e os azimutes. Os elementos de curva horizontal circular e espiral também podem ser computados e desenhados pelo complemento para serem recalculados e inseridos a uma tabela de horizontais já existente. Se forem adicionados pelo complemento esses dados serão armazenados na tabela “CUVAS\_SIMPLES” do banco de dados, para que possam ser alterados pelo usuário. O desenho final é armazenado em um arquivo no formato *geopackage* para que se possa usar o mesmo algoritmo de conversão em tabela de estacas.

Tabela 2: Exemplo de tabela de atributos para uma *layer* de traçado horizontal usada pelo complemento no formato *geopackage*. Todas as unidades estão em metros e graus.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Atributo | Tipo | Descrição | Raio | Angulo de Deflexão (Delta) | Tangente Externa (T) | Desenvolvimento (D) |
| 1 | T |  |  |  |  |  |
| 2 | E |  | 520 | 16,67 | 128,073 | 120 |
| 3 | C |  | 520 | 16,67 | 128,073 | 120 |
| 4 | E |  | 520 | 16,67 | 128,073 | 120 |
| 5 | T |  |  |  |  |  |
| 6 | E |  | 400 | 9,382 | 239,073 | 200 |
| 7 | C |  | 400 | 9,382 | 239,073 | 200 |
| 8 | E |  | 400 | 9,382 | 239,073 | 200 |
| 9 | T |  |  |  |  |  |
| 10 | E |  | 683,1 | 33,56 | 201,235 | 180 |
| 11 | C |  | 683,1 | 33,56 | 201,235 | 180 |
| 12 | E |  | 683,1 | 33,56 | 201,235 | 180 |

Todo esse processo é representado pelo fluxograma da Figura 4 (Anexo A). Destaca-se o processo iterativo feito para cada atributo contido na *layer vetorial*, o que resulta em um elemento geométrico isolado para cada atributo ou, colocando de outra forma, para cada linha na Tabela 2, do qual se faz a segmentação em estacas. Essas estacas são armazenadas na tabela “ESTACA” do banco de dados dentro do arquivo compactado de projeto.

**3****.3 Perfil e curvas verticais**

Para a parte da sessão vertical de um projeto de estradas o complemento desenvolvido realiza inicialmente a tradução do perfil horizontal em um perfil longitudinal do relevo do terreno onde possa ser traçado, pelo usuário, por meio de uma interface simples e intuitiva, o greide e definidas as curvas verticais. Esse processo é mostrado pelo fluxograma na Figura 6 (Anexo A).

Para cada estaca criada no passo anterior deve-se extrair uma elevação, relativa às suas coordenadas, a partir da qual o terreno para a elaboração do perfil vertical será definido. Esse terreno é basicamente a vista de perfil da sessão horizontal e sua utilidade é servir de base para o greide de projeto. Essas elevações são obtidas através de *rasters* e podem ser opcionalmente interpolados por um algoritmo que foi construído especificamente para o complemento. A partir das coordenadas da estaca localiza-se o pixel no qual esse ponto está contido (ponto em vermelho na Figura 5 - Anexo A). Existem 8 pixels ao redor do central e são escolhidos os três com os centros mais próximos do ponto em questão. A elevação interpolada é calculada com as equações 1 e 2:

(1)

(2)

Onde D é o comprimento da diagonal (Figura 5 - Anexo A), P os pesos da média, v a elevação de cada pixel associado, L a distância do ponto ao centro do pixel central e L1, L2 e L3 as distâncias aos centros dos pixels mais próximos. Vale ressaltar que os comprimentos a e b de cada pixel podem ser convertidas para metros para que esse processo seja feito e o valor de v é proporcional ao valor do pixel, que é o módulo de um vetor tridimensional RGB. Os valores da elevação associados às estacas são armazenados na tabela “RELEVO\_SESSÃO”.

Para uma melhor interação com usuário foi necessário implementar uma interface de desenho e edição do greide sobre o relevo. Como a parte de perfis expandiria as classes atuais do projeto além do limite considerado legível, optou-se por isolar suas funcionalidades em uma classe isolada denominada “Ui\_Perfil” que cuida da interface de edição interativa do greide buscando permitir ao usuário o desenho sobre o relevo feita a partir da *pyqtgraph*. Essa mesma interface serviu de base para a sessão transversal e o Diagrama de Bruckner. Procurou-se manter o mesmo padrão de entidades e adicionar subclasses da biblioteca *pyqtgraph* com destaque a subclasses ROI. Esta subclasse é útil para os desenhos com as adicionais e métodos para determinação da posição dos vértices e com métodos carregados com sinais Qt que abrem os diálogos de edição de vértices e rampas.

A subclasse “CustomViewBox” foi criada para tornar a movimentação e comandos de arrastar do mouse mais familiares para usuários de softwares CAD. Destaca-se que a classe “Perfil\_cv” encapsula os cálculos básicos para determinar as cotas em cada ponto da curva vertical dado uma progressiva e a inclinação das rampas adjacentes ao vértice. Na Figura 7 (Anexo A) apresenta-se o diagrama de classes da interface gráfica para o perfil. Essas classes possuem também os métodos onde são realizadas todas operações relacionadas à manipulação dos vértices, salvamento, restauração e cálculos auxiliares tais como a determinação da inclinação de rampas, comprimentos e cotas.

Passada a fase de modelagem, o passo seguinte foi aplicação dos conceitos na plataforma através de código, sempre seguindo os padrões da plataforma. A primeira parte foi o desenvolvimento das interfaces utilizando o *framework* Qt seguindo um padrão clássico, buscando ser o mais próximo do utilizado pelo sistema com o uso extensivo do sistema de *signals e slots* para a intercomunicação entre as classes do perfil e Estacas. O *pyqt5* facilitou esse processo o que pode tornar a execução mais reativa e assíncrona, o que também foi usado para que o programa apresente quaisquer alterações de valores refletidas nos desenhos em tempo real.

**3****.4 Perfil transversal** **e Diagrama de Bruckner**

Antes da definição do perfil transversal é necessário combinar a tabela de estacas horizontais com as verticais, já que as estacas intermediárias de curvas horizontais e de curvas verticais nem sempre irão coincidir ou serão estacas inteiras. Esse processo é feito organizando as duas tabelas em ordem de progressivas, que é a distância horizontal acumulada, somando-se as duas listas de estacas, eliminando as repetições e reorganizando em ordem crescente de estacas. O resultado final é uma tabela de interseção entre o perfil vertical e o horizontal.

Para cada estaca na tabela de interseção calcula-se uma coordenada e calcula-se qual é a reta perpendicular ao ponto dado o ângulo de azimute da estaca. Essa reta é o que define a sessão transversal para aquela estaca e, em um procedimento semelhante ao da obtenção das elevações para o traçado horizontal, calcula-se todas as cotas ao longo desse reta. A largura a ser considerada e precisão são configuráveis.

Foi criada uma interface semelhante à do perfil vertical para definir a sessão tipo que representa o corte transversal da pista que deseja-se projetar. Essa interface permite o usuário definir os taludes de corte e aterro, valetas e a pista em si. Aqui é utilizada a subclasse “ssROI” que gerencia os vértices editáveis para a sessão tipo. Todo esse algorítimo está apresentado no fluxograma da Figura 8 (Anexo A).

O próximo passo é calcular a interseção entre a sessão tipo definida pelo usuário e o terreno. Optou-se por criar uma biblioteca própria para esse processamento e para os cálculos de volume ao invés de usar uma solução já reconhecida como *scipy* ou a própria API do QGIS. Essa escolha se justifica pela carência dessas outras opções no cálculo de interseção entre curvas e outras funções que geralmente se encontram em núcleos CAD. Para a criação dessa biblioteca procurou separar os problemas em partes simples, seguindo a orientação a objetos, iniciando-se com a definição de um ponto. Depois uma reta a partir de dois pontos e limitada por eles, definindo-se métodos para a obtenção do ponto de interseção. Com uma lista de retas interligadas definiu-se a classe curva, com métodos para a interseção com outras curvas. A partir de duas curvas se define uma face, desde que essas curvas possuam mais de duas interseções, com métodos para o cálculo de sua área e por fim um prismoide a partir de múltiplas faces.

As áreas das faces são determinadas pela fórmula de Gauss, já que todas as faces são perpendiculares ao traçado horizontal, da seguinte maneira:

(3)

Onde xi e yi são as coordenadas do i-ésimo pixel em um polígono e A a área total desse polígono. O cálculo de volume do prismoide se dá a partir do somatório do produto das distâncias entre as faces e as médias de suas áreas. Esse método é conhecido como semissoma. O objeto prismoide pode ser armazenado diretamente no disco a partir da biblioteca *pickle*, que permite serializar um objeto python armazenando todos seus atributos de maneira que ele possa ser recarregado na memória e instanciado novamente. Porém essa solução pode falhar e então o complemento irá recalcular novamente o prismoide do traçado a partir dos dados no banco de dados. Os relevos são armazenados na tabela “RELEVO\_SESSAO” e as sessão tipo em “TRANSVERSAL”.

A partir desse prismoide pode-se calcular o volume total de aterro e de corte para o traçado, e gerar o Diagrama de Bruckner que se trata dos volumes acumulados ao longo do traçado. Também foi criada uma interface baseada na “Ui\_Perfil” para visualizar esse diagrama em um certo intervalo de estacas e para fazer medições de distâncias e volume sobre ele. Para uma melhor visualização dos volumes adiciona-se 1000 m³ ao menor volume do gráfico. A interface também permite a determinação da onda de Bruckner sendo esta definida pelo retângulo de mesma área que a área abaixo do trecho delimitado do diagrama de massas, dado que o usuário escolha a base desse retângulo que é a distância média de transporte. Essa ferramenta é útil para que seja feito o balanço de volumes a partir do cálculo do momento de transporte.

**4****. RESULTADOS**

O projeto está sendo mantido publicamente no *github* e pode ser visualizado através do endereço: <https://github.com/matheusfillipe/Topografia> onde também se encontram instruções para a instalação. A intenção é que também esteja disponível no repositório do padrão do QGIS. A licença escolhida para o software foi a GPL versão 2.0, que é a mesma licença de código aberto utilizada pelo QGIS.

A árvore de arquivos final do código fonte (Figura 9 - Anexo A) se baseia em três principais diretórios. Primeiramente “view” contém a lógica da interface gráfica. Por exemplo o arquivo “config.py” contém a classe para a janela de configurações e “estacas.py” os vários diálogos com as tabelas de estacas. Os layouts das janelas e diálogos foi modelada usando o formato *Designer ui* que pode ser editado com o software Qt Designer. A pasta “model” contém a interface entre os dados na memória e o banco de dados e armazenamento. A pasta “controller” contém os algoritmos descritos na sessão anterior. Alguns destaques são os diretórios “controler/Geometria”, que contém a biblioteca para cálculo de interseções de curvas e cálculos relacionados ao prismoide, e o arquivo “model/utils.py” que contém várias utilidades relacionadas ao uso da API do QGIS dentre elas a função de interpolação de rasters.

O complementose apresenta em estado funcional, cobrindo o básico do projeto geométrico de estradas e permitindo a extração dos desenhos em formato CAD, de maneira simplificado, e das tabelas em *csv* (Planilha com valores separados por vírgula)*.* A seguir será apresentado de forma resumida o funcionamento do complemento mostrando suas capacidades abordando um projeto simples dentro do QGIS.

**4.1 O fluxo de projeto com o complemento**

Na Figura 10 (Anexo A) é apresentada a interface principal do complemento na barra de ferramentas do QGIS, sendo composta por seis ícones, sendo eles em ordem partindo da esquerda: criação do arquivo de projeto, abertura de arquivo um arquivo de projeto (.zip), salvar o projeto atual, configurações de projeto, mostrar imagens de satélite e o gestor dos arquivos de projetos.

Para criação, abertura e armazenamento de arquivos topográficos foi utilizada a interface padrão do sistema operacional no que se refere à seleção de arquivos e pastas, com a implementação específica feita internamente, o que facilita o uso do sistema. Para criação de arquivos topográficos, o usuário apenas irá escolher o local e o nome a qual serão salvas as configurações do projeto. Este poderá ser aberto com a mesma lógica, e quando isso ocorrer todas as configurações e traçados estarão importados para o QGIS, não precisando armazenar em arquivos separados. Para seleção dos arquivos topográficos do tipo *shapefile* (.shp), também foi utilizada a interface padrão com a diferença que é escolhido a pasta e não os arquivos separados.

Vale ressaltar a necessidade de se selecionar o sistema de projeção cartográfica adequado nas configurações de projeto do QGIS e nas layers importadas antes que se inicie qualquer procedimento usando o complemento. O primeiro passo é criar um novo arquivo topográfico por meio do primeiro ícone. O segundo passo seria fazer as configurações básicas do projeto através do quarto ícone. Apesar de existir o botão de salvar (terceiro ícone), todas as modificações são salvas no momento em que editadas e aplicadas dentro do complemento, exceto pelas janelas de perfil vertical, transversal e diagrama de Bruckner, que são implementadas com a *pyqtgraph.*

Todas configurações do projeto geométrico são realizadas na janela de configurações (Figura 11 - Anexo A). Nela pode ser configurada a classe de projeto que influencia nas inclinações limite do greide e velocidade que são mostrados abaixo. A velocidade é por padrão para terreno ondulado, mas todos essas inclinações para os três tipos de terreno e a velocidade podem ser editados independentemente da classe de projeto caso seja necessário. Outros valores que podem ser interessantes nessa janela são o *checkbox* interpolar rasters, que se verificado perfomará a interpolação dos pixels nas operações de extração de elevações; a distância entre as estacas, que é por padrão de 20 metros; EMAX que é a máxima taxa de superelevação admissível em m/m, utilizada para calcular o raio mínimo recomendado em curvas horizontais; o separador CSV utilizado nas planilhas a serem exportadas e as configurações para a extração de elevação nas sessões transversais, que são a largura transversal total de cada lado a partir do eixo, medido em metros, e o número de pontos a serem distribuídos igualitariamente ao longo dessa largura, de onde se extraí valores de elevação. Esse último valor deve ser mantido como mínimo possível para que os procedimentos com a sessão transversal carreguem mais rápido.

A partir das configurações realizadas, o próximo passo é a criação dos traçados e desenvolvimento das curvas horizontais. Qualquer layer vetorial que pode ser criada no QGIS ou extraída de um programa CAD pode ser utilizada. O quinto ícone da barra de ferramentas carrega um *tilemap* com as imagens de satélite e imagens de elevação do terreno, fornecidas pela Google, com o intuito de auxiliar na definição do traçado. Também existe um botão que serve como um atalho para gerar a layer com o traçado horizontal dentro do QGIS (Figura 12 - Anexo A). Quando a layer estiver definida (Figura 13 - Anexo A), para criar uma tabela de estacas basta usar o botão “Novo arquivo de estacas” da janela da Figura 12 (Anexo A) e selecionar a layer com o traçado. A tabela aparecerá já calculada como na Figura 14 (Anexo A). Ressalta-se que múltiplos traçados podem ser definidos dentro de um mesmo arquivo topográfico e eles aparecerão em uma lista dentro da janela de gestão de traçados (Figura 15 - Anexo A), onde eles podem ser abertos, excluídos ou duplicados.

As curvas horizontais podem ser criadas a partir do botão curvas (Figura 14 - Anexo A). O complemento irá criar uma layer interna do traçado com curvas e mostra-la. A janela de edição de curvas aparecerá conforme a Figura 16 (Anexo A). Nessa interface podem ser definidas dois tipos de curvas: circular simples e circular com transição simétrica. Os principais parâmetros das curvas podem ser editados e a curva pode ser armazenada com o botão “salvar”. O botão “Gerar Todas” gera todas as curvas para cada ponto de interseção usando os valores mínimos recomendados para o desenvolvimento (D) arredondados para cima e sendo múltiplos da distância entre estacas. O último passo é dado com o botão “Calcular” que importa a layer modificada e recalcula a tabela de horizontais.

De posse do traçado horizontal é necessário calcular os dados de elevação para cada estaca por meio de um arquivo *geotiff*. Isso é feito a partir do botão “Obter Cotas GeoTiff” (Figura14) onde deve-se selecionar esse arquivo. A coluna cotas da tabela será preenchida.

A partir do traçado horizontal e suas respectivas cotas, o complemento permite a criação do perfil longitudinal e as curvas verticais por meio do botão “perfil de trecho”. Por padrão é traçado um greide que segue a mediana do terreno, os vértices das extremidades têm suas posições horizontais fixadas de acordo com o início e término do perfil horizontal e o terreno também está plotado. Conforme pode ser observado na Figura 17 (Anexo A) os círculos azuis representam as estacas, com o intuito de facilitar a determinação do traçado vertical ideal. O greide pode ser editado adicionando vértices com o clique do botão esquerdo do mouse e arrastando-os para a posição desejada. A rampa de maior inclinação do trecho fica sempre sublinhada de vermelho e a velocidade recomendada para essa inclinação mostrada na parte superior.

Podem ser inseridos valores para as rampas quando clicadas com o botão direito. As curvas podem ser inseridas ao se clicar com o botão direito sobre um vértice e acessar a “opção editar vértice” conforme mostra Figura 18. Essa interface apresenta todos os parâmetros da curva e das rampas adjacentes. “L utilizado” é o valor do comprimento da curva em metros que pode ser editado e visualizado em tempo real assim como na interface de curvas horizontais. Nessa interface a escala do desenho pode ser alterado ao se arrastar o mouse segurando o botão direito. O botão zoom retorna a imagem ao zoom e escala originais a qual inserem todo o trecho na tela. O último passo é usar o botão “Aplicar” ou “Fechar” e salvar o perfil longitudinal.

Em seguida pode-se retornar à janela da Figura 14 (Anexo A) e usar o botão “Abrir verticais” que irá calcular a tabela de verticais e abrir a janela da Figura 19 (Anexo A). Nessa janela pode ser calculada também a tabela de interseções, necessária para definir a sessão transversal que é acessada através do botão “Definir sessão tipo” através da janela mostrada pela Figura 21. Na primeira vez o complemento irá calcular as sessões do terreno para cada estaca. As sessões tipo podem ser aplicadas para diferentes trechos de estacas pelo botão “Aplicar”. Elas também podem ser extraídas ou importadas usando o formato dxf com os botões “Importar” e “Exportar”. O volume total de corte e aterro pode ser calculado com o botão “Volume”.

De volta à janela da Figura 19 (Anexo A), o diagrama de Bruckner pode ser calculado e o resultado mostra é apresentado na Figura 24 (Anexo A) com uma interface semelhante à da sessão transversal e perfil vertical. Pode ser selecionado o intervalo de estacas de interesse para calcular uma curva de Bruckner com a distância de transporte editável conforme apresentado na Figura 25 (Anexo A).

**4.2 Validação do complemento**

Os resultados das tabelas e do cálculo de volumes foram comparados para um mesmo traçado, greide e sessão transversal entre o complemento e o Bentley TOPOgraph. Os resultados das tabelas foram idênticos.

Os testes realizados com o complemento demonstraram que o mesmo configura um grande avanço na área, pois permite a criação de traçados utilizando como referência arquivos vetoriais disponibilizados pelo IBGE e imagens de satélite. Esse fato agiliza o trabalho, pois a partir das imagens têm-se uma noção mais fidedigna dos usos do solo existentes na área de projeto.

Ressalta-se que o complemento foi aplicado pelos estudantes da disciplina de projeto geométrico de estradas no período 2019 II, do curso de Engenharia Civil. Foi proposto à eles que apliquem na prática o complemento validando os cálculos e indicando melhorias. Pretende-se, com isso, a partir da visão dos usuários, melhorar o complemento.

**5****. CONCLUSÃO**

Pode-se dizer que esse projeto tem sua originalidade e pionerismo com relação ao desenvolvimento de softwares desta natureza sob a licença livre com funcionalidades semelhantes aos pagos. Com relação à interface, as mesmas foram inspiradas a partir dos sistemas mais utilizados da área como o Topograph, buscando-se assim ter uma curva de aprendizagem mais rápida. Também buscou-se inovar em alguns aspectos que agilizem o processo de projeto por meio de interfaces mais enxutas e com capacidade de resposta mais aguçada. Tudo com o intuito de promover um fluxo de trabalho o mais direto possível, sem perder a capacidade de correções, mas evitando resservidos.

O complemento possui código seguindo os padrões mais utilizados na engenharia de software no mercado, facilitando a manutenção e inclusão de novas funcionalidades sob a demanda necessária. Ressalta-se que devido à linguagem python ser de alto nível e interpretada, alguns processos tais como o cálculo da sessão transversal e os cálculos volumétricos tem uma certa lentidão no processamento devido sua profundidade de iteração.

Como é de costume quando se trata de software nunca há um verdadeiro fim do desenvolvimento já que novas funcionalidades sempre podem ser adicionadas. A expectativa é que o projeto continue a ser desenvolvido e que novas funções como o cálculo de superelevação e superlargura seja possível em próximas versões.

**6****. REFERÊNCIAS BIBLIOGRÁFICAS**

CAMPOS NETO, S.; PAMBOUKIAN, S. V. D.; CRAVEIRO, M. V.; BARROS, E. A. R. Estudo comparativo de ferramentas computacionais que utilizam tecnologia BIM para desenvolvimento de projetos de engenharia civil. In: XI Congresso Brasileiro de Educação em Engenharia, Belém. **Anais...** Pará, 2012.

CNT - CONFEDERAÇÃO NACIONAL DO TRANSPORTE. **Boletim Estatístico.** Brasília, 2018. Disponível em: <http://www.cnt.org.br/Boletim/boletim-estatistico-cnt>. Acesso em: 17 de maio 2018.

DNIT - DEPARTAMENTO NACIONAL DE INFRA-ESTRUTURA DE TRANSPORTES. **Diretrizes básicas para elaboração de estudos e projetos rodoviários escopos básicos/instruções de serviço.** Brasília, 2006. Disponível em < http://www1.dnit.gov.br/arquivos\_internet/ipr/ipr\_new/manuais/diretrizes\_basicas\_instrucoes\_servicos.pdf>. Acesso em: 11 nov. 2015.

GUIA LIVRE. **Referência de Migração para Software Livre do Governo Federal**. Organizado por Grupo de Trabalho Migração para Software Livre. Brasília, 2005.

MAIA, R. O.; SILVA, F. A.; PAZOTI, M. A.; ALMEIDA, L. L.; PEREIRA, D. R. **Desenvolvimento de um dispositivo para apoio ao ensino de computação e robótica.** Revista Colloquium Exactarum, v. 6, 2014.

MANGHI, G.; CAVALLINI, P.; NEVES, V. **Quantum GIS: Um desktop potente e amigável.** Revista FOSSGIS, v. 2, 2011.

MENEZES, R. R. V.; SANTOS, A. P.; CARVALHO, C. A. B.; FERRAZ, A. S. **Emprego dos softwares TCP-MDT e topoGRAPH para projeto geométrico de estrada.** In: V Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação, Recife. **Anais...** Pernambuco, 2014.

OLIVEIRA JUNIOR, M. G. **Projeto geométrico de vias com modelagem digital do terreno em ambiente de software livre e sistema de informações geográficas**. 2007. 186 f.. Dissertação (Mestrado em Engenharia de Transportes) - Instituto Militar de Engenharia, 2007.

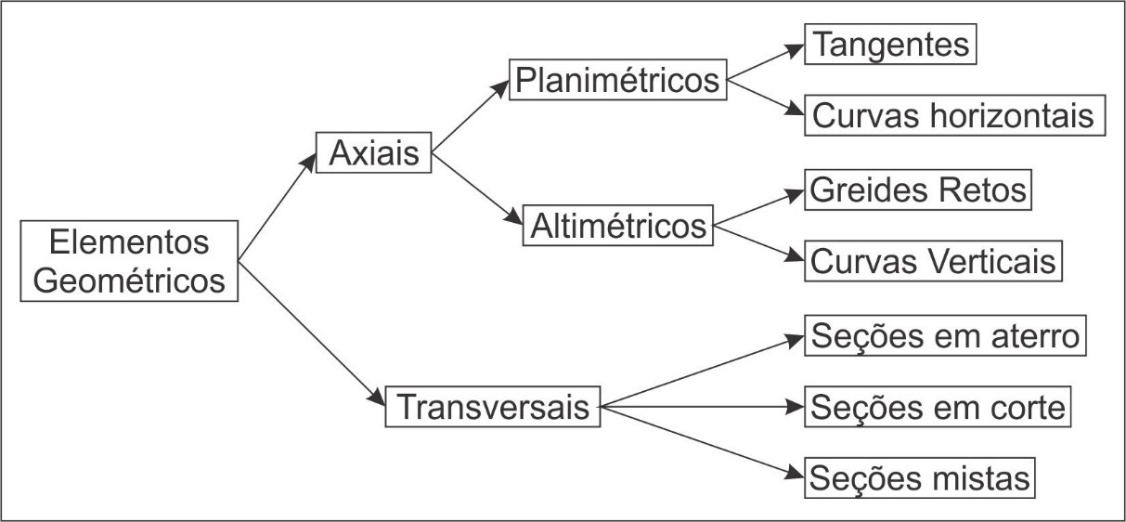
OSGEO (org.). **Descubra o QGIS**: QGIS - A liderança do SIG de código aberto. [*S. l.*], 20--. Disponível em: https://QGIS.org/pt\_BR/site/about/index.html. Acesso em: 11 nov. 2019.

PONTES FILHO, G. **Estrada de rodagem - projeto geométrico.** São Paulo, Editora Eletrônica, 1998.

SILVA JÚNIOR, J. A. D. **Metodologia para a utilização de dados topográficos em projetos geométricos automatizados de vias**. 2003. 144 f.. Dissertação (Mestrado em Engenharia de Transportes) - Instituto Militar de Engenharia, 2003.

VASCONCELOS, G. M.q & BASSO, L. **Sistemas de Transportes Brasileiros: Origens, Estado Atual e Desafios.** Revista Jovens Pesuisadores Ano V, v. 8, 2008.

ANEXO A - FIGURAS

  
Figura 1: Elementos geométricos de uma estrada.

Fonte: Pontes Filho (1998, p. 30).

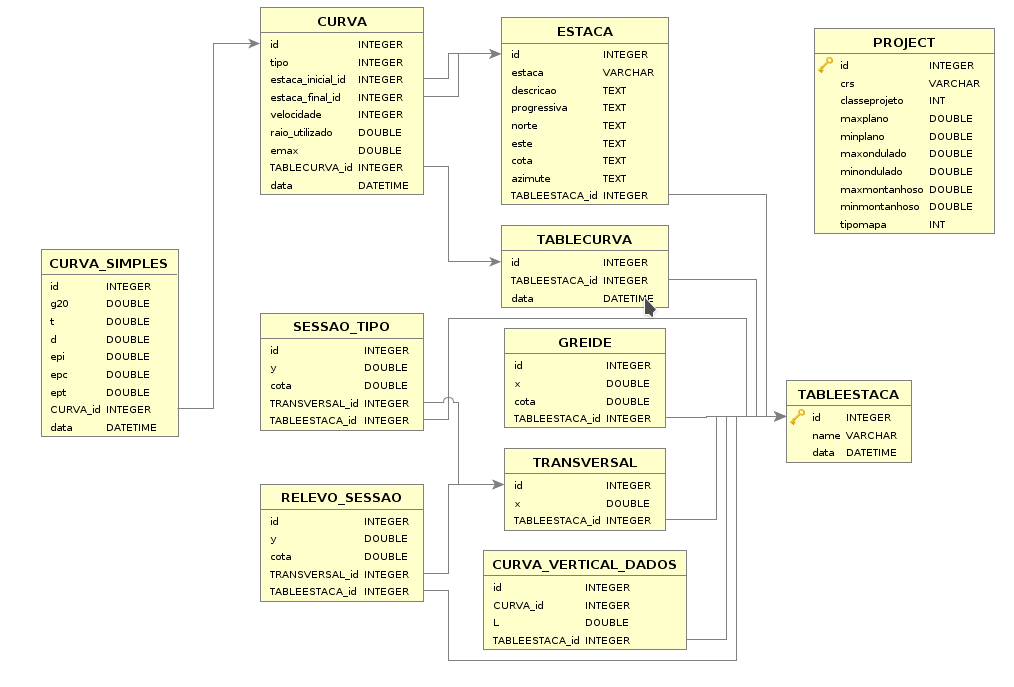


Figura 2: Diagrama entidade relacionamento para os tipos de dados do complemento.

  
Figura 3: Árvore de arquivos dentro do arquivo compactado de projeto.

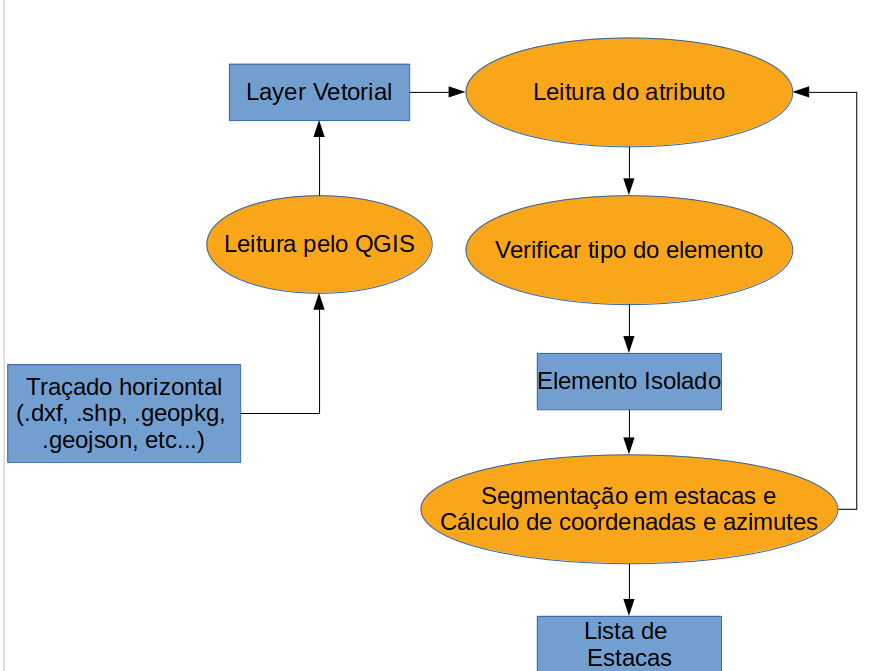


Figura 4: Algoritmo de segmentação do traçado horizontal a partir de um desenho

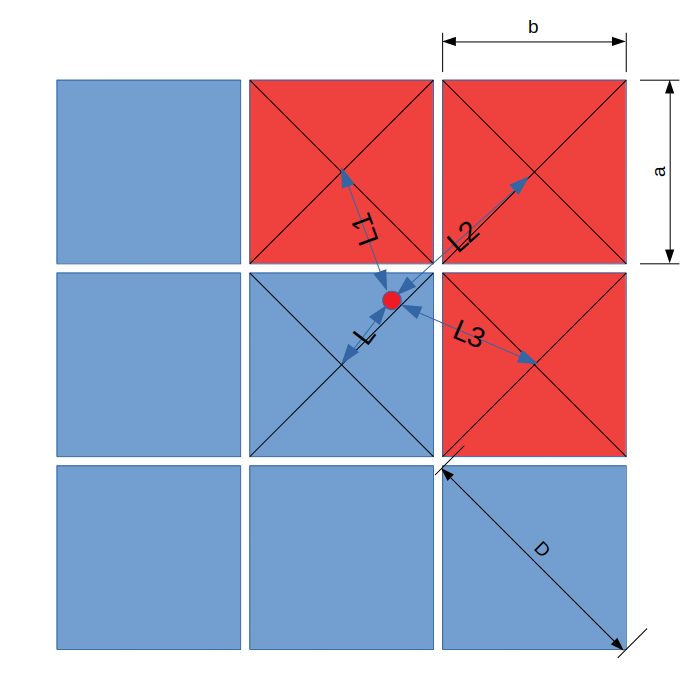
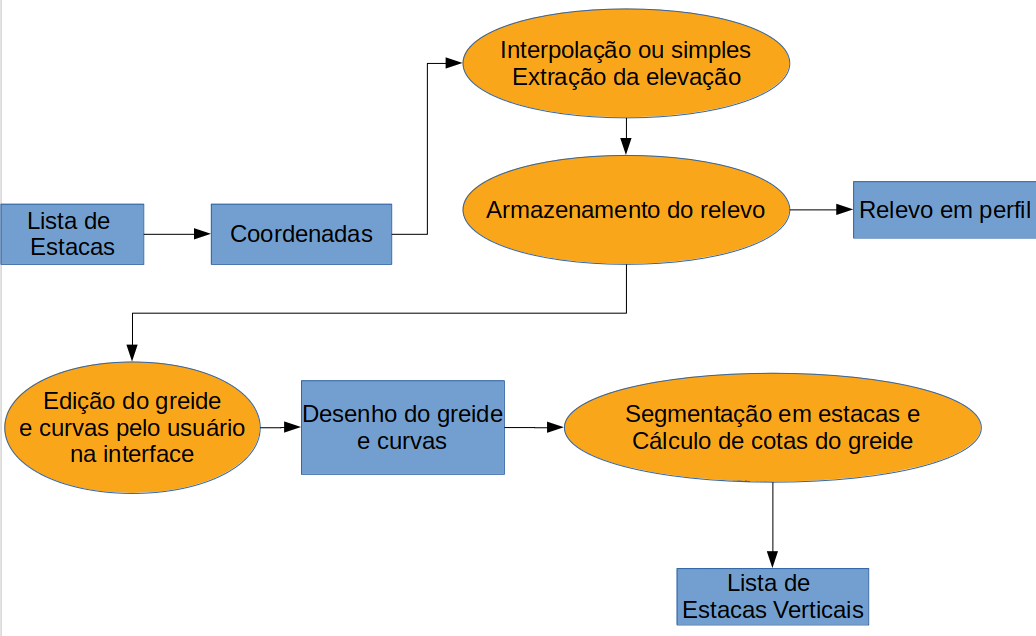
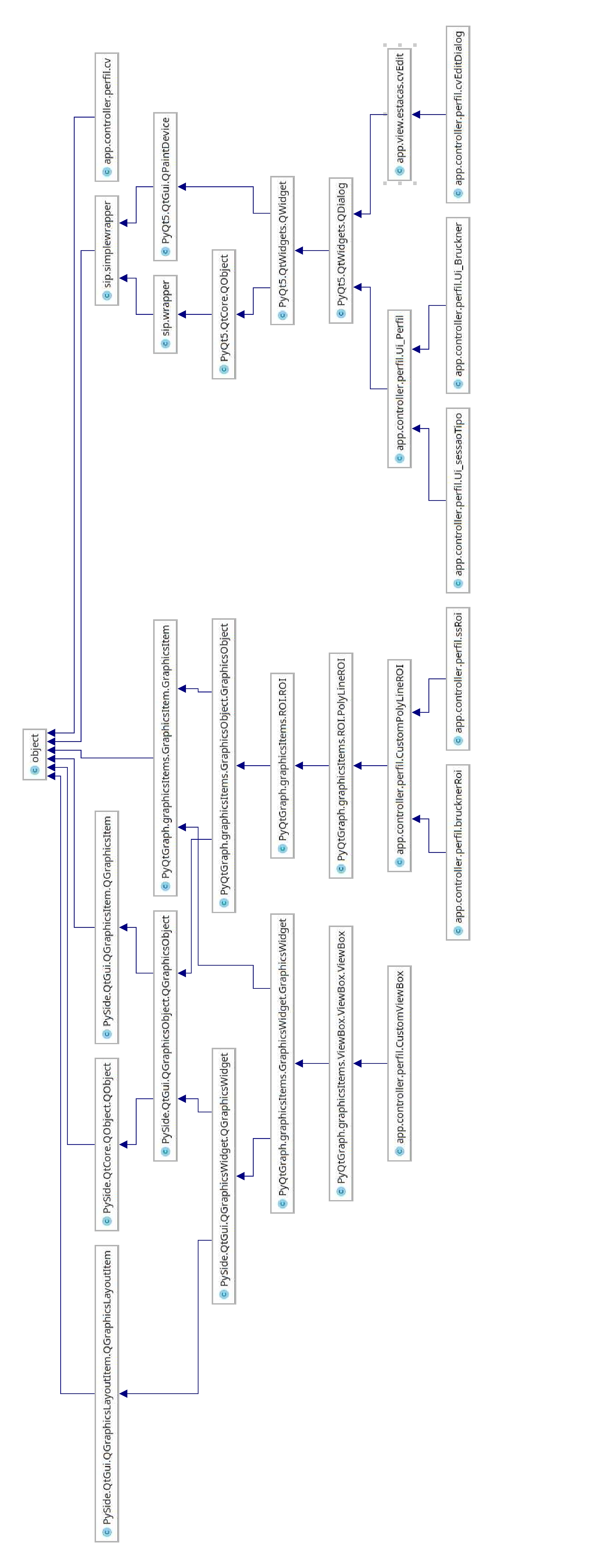


Figura 5: Interpolação de *rasters*. Cada quadro representa um pixel. Os pixeis em vermelho são os escolhidos para a interpolação da elevação do ponto em vermelho.

  
Figura 6: Processo de definição do perfil vertical.

  
Figura 7: Diagrama de classes do algoritmo de visualização e cálculo de perfis.

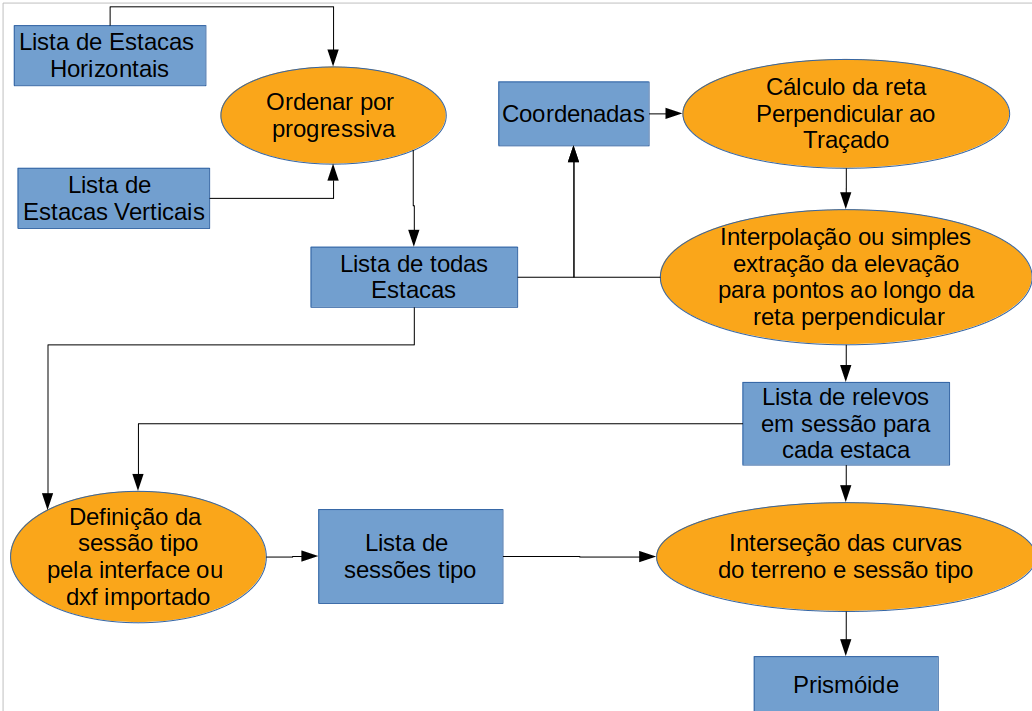


Figura 8: Combinação das estacas verticais e horizontais na tabela de interseções para a definição das sessões transversais.

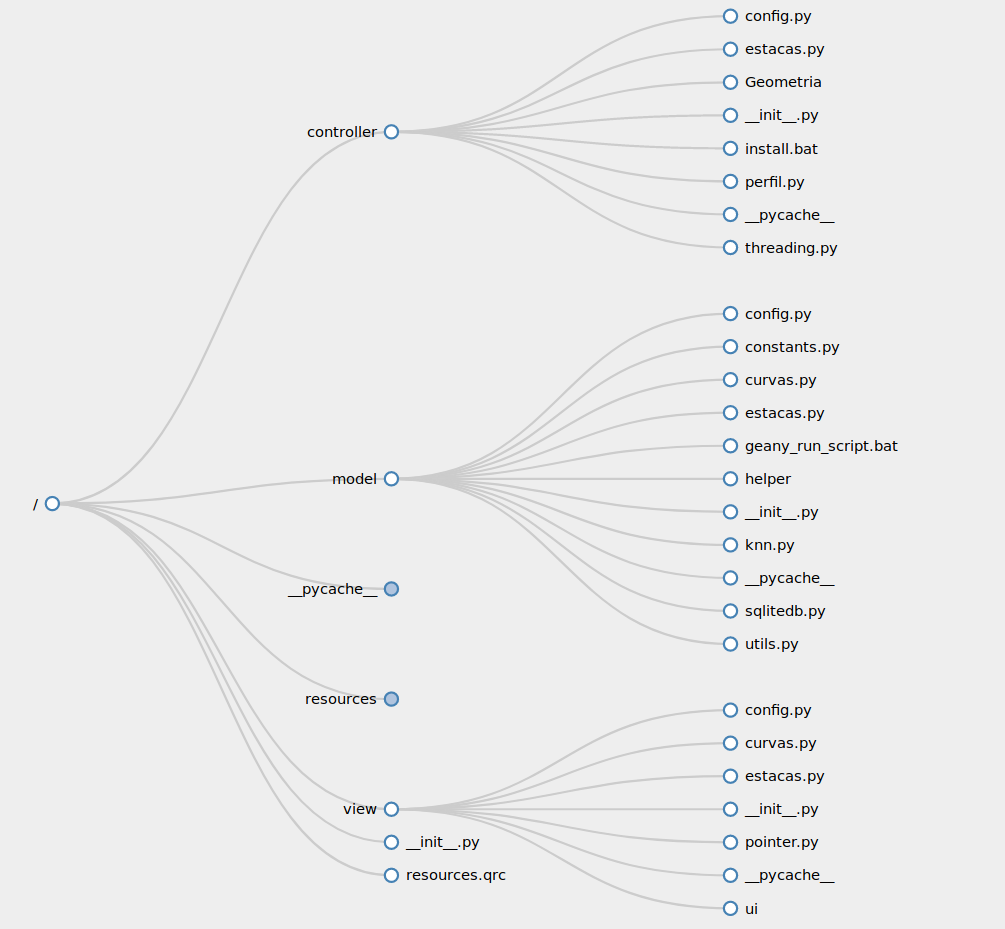
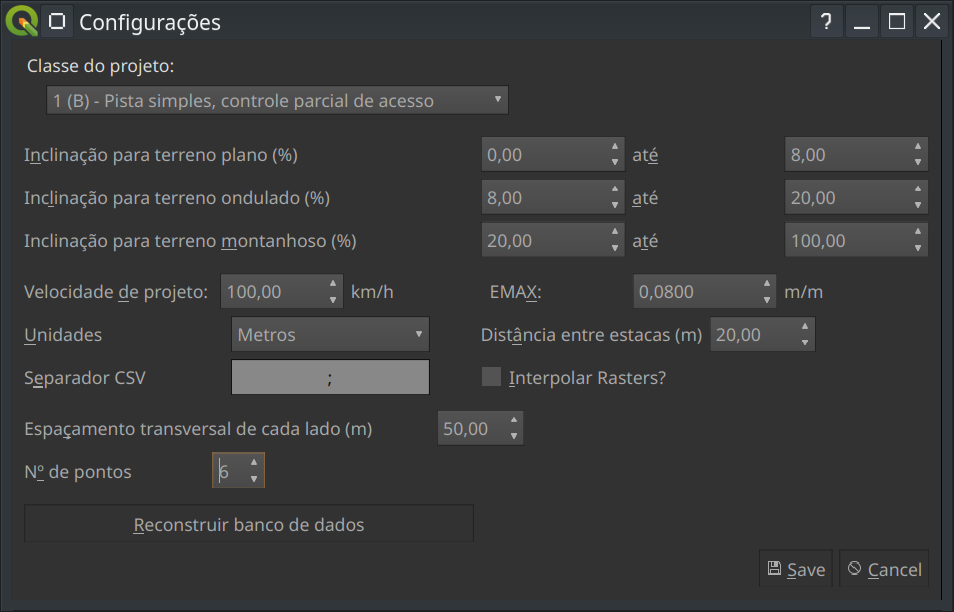
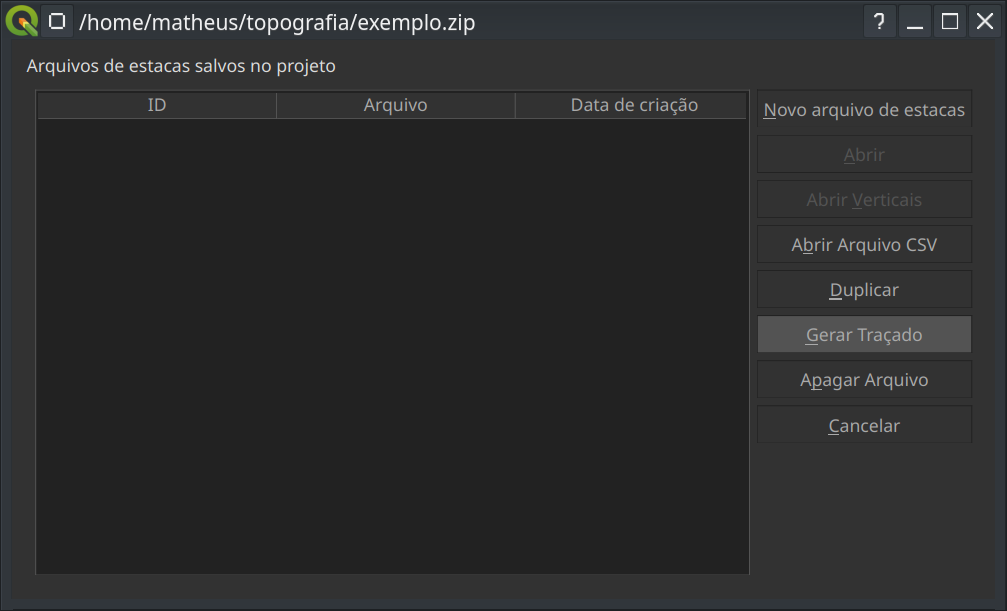
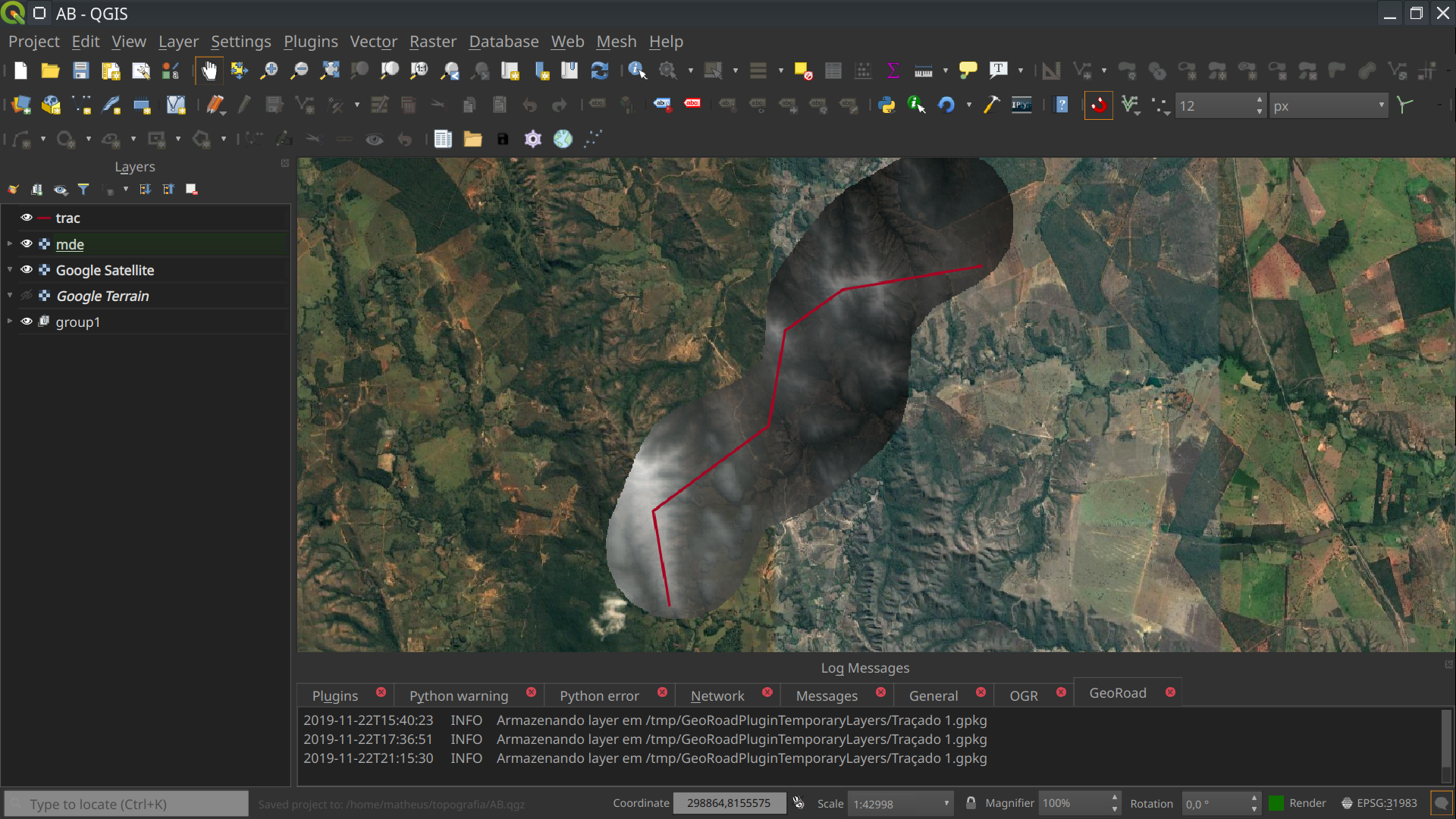
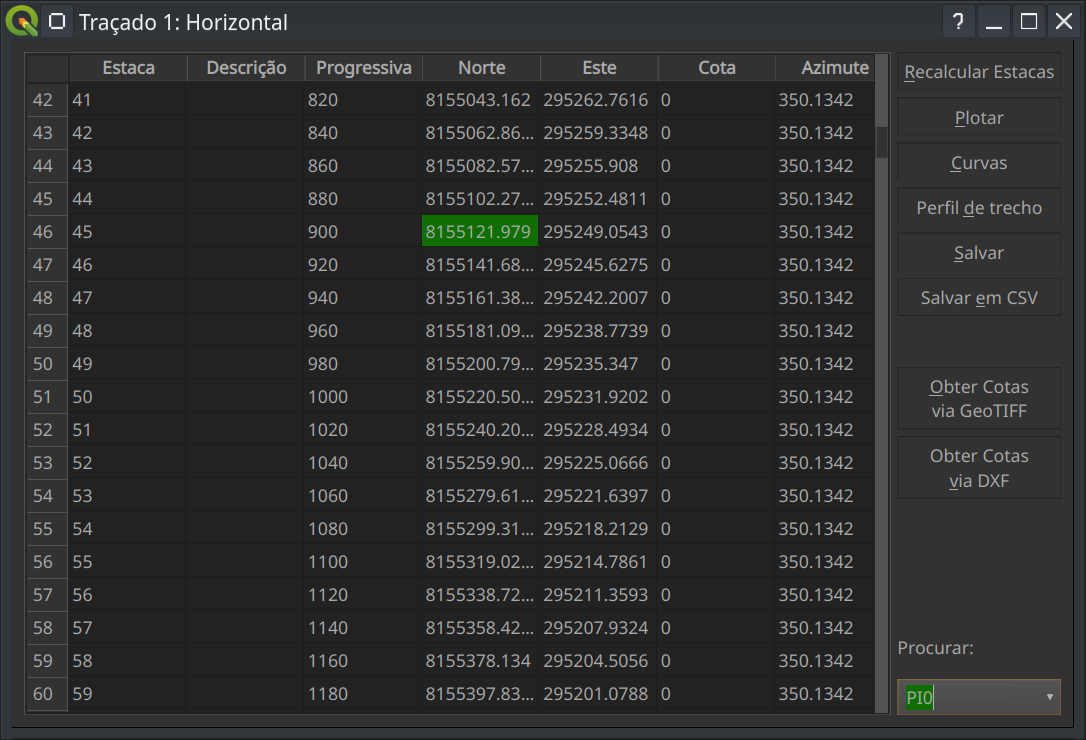
  
Figura 9: Árvore de arquivos do código fonte.

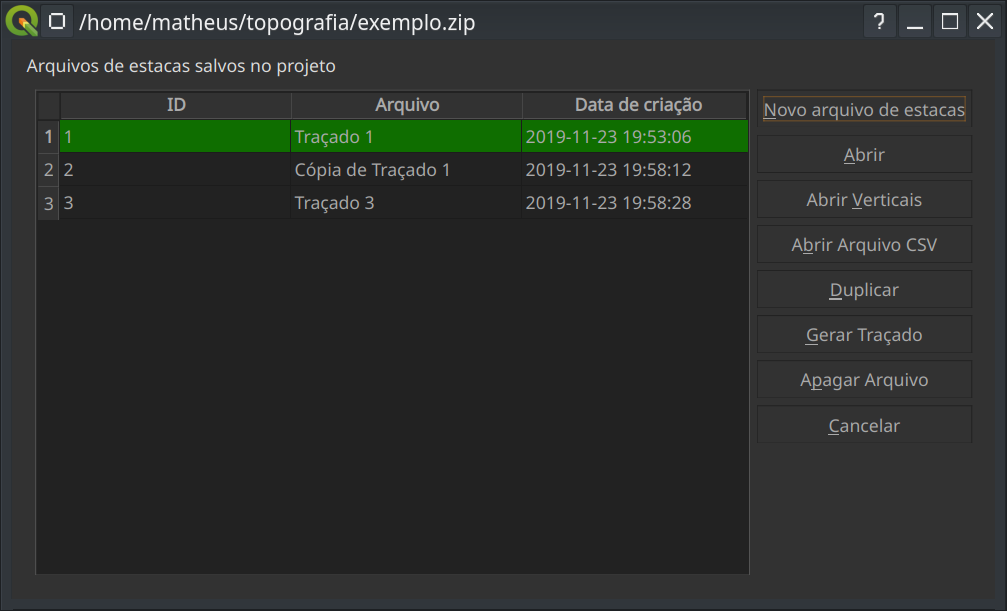
  
Figura 10: Interface e acesso do complemento desenvolvido dentro da barra de ferramentas do QGIS.

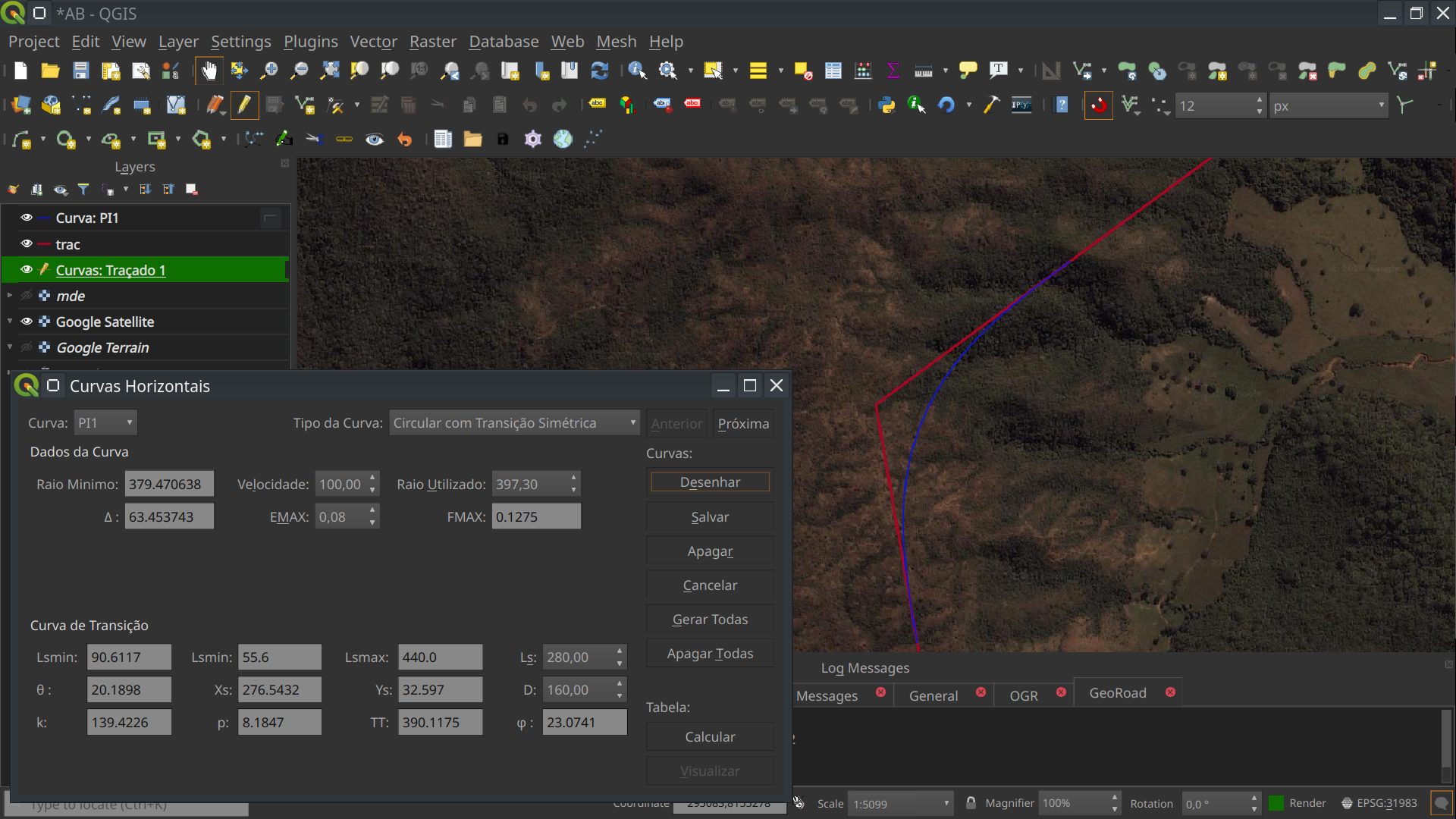
  
Figura 11: Janela de configurações do projeto.

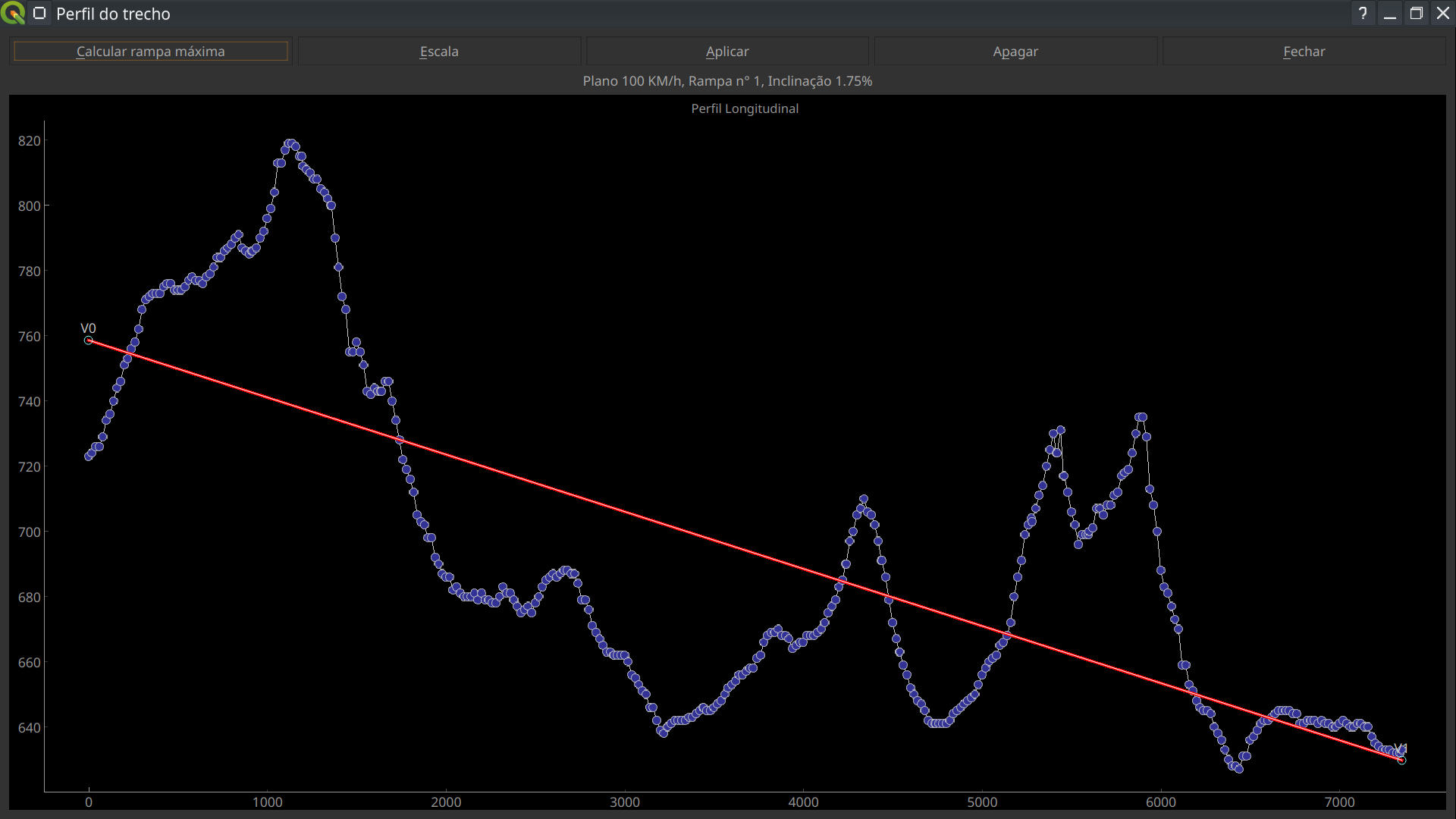
  
Figura 12: Janela para manipulação de traçados. O botão destacado é um atalho para criar uma nova layer no QGIS que irá servir como traçado horizontal.

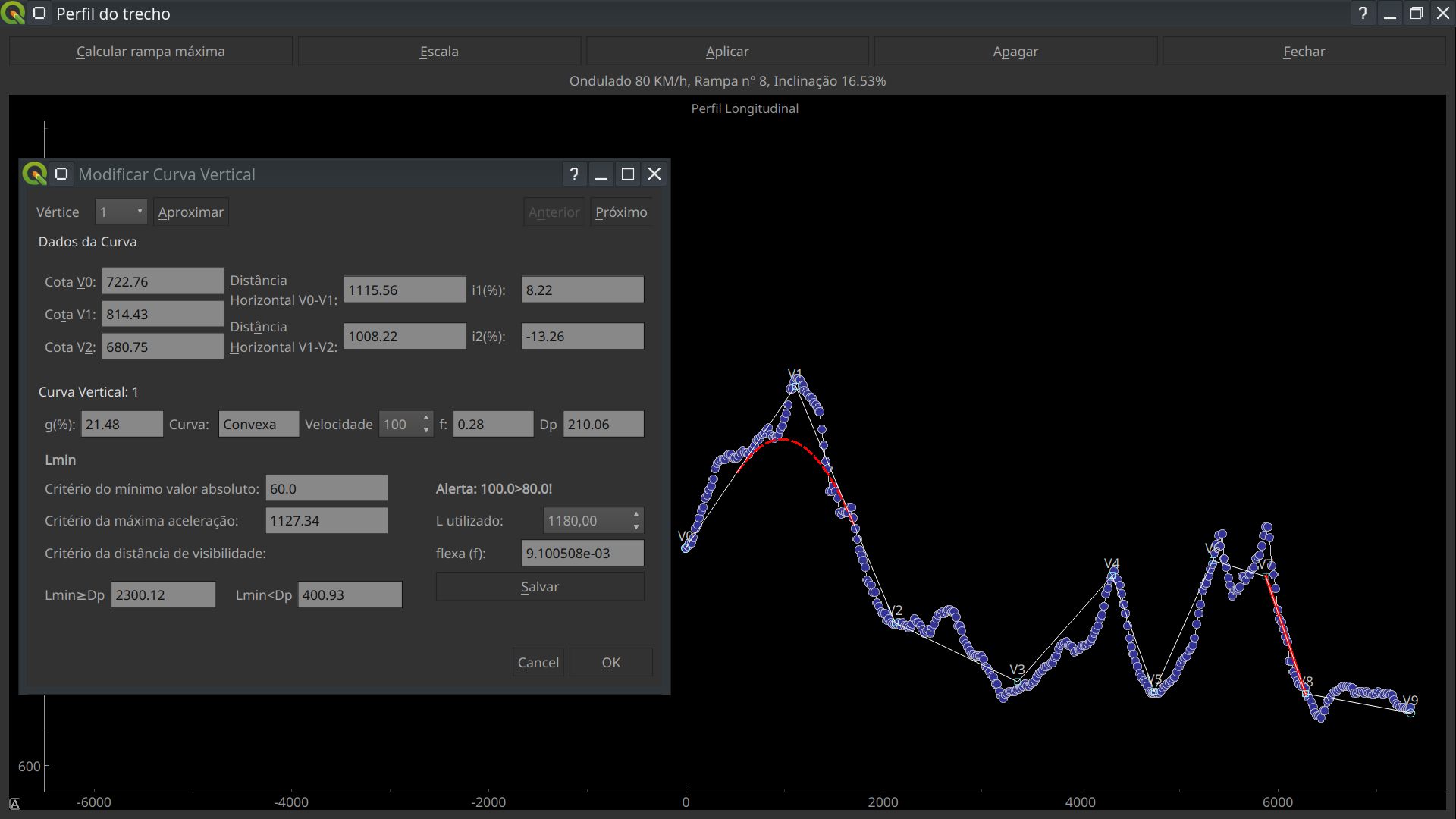
  
Figura 13: Exemplo meramente ilustrativo de um traçado horizontal com as tangentes que um usuário pode importar como uma layer do QGIS.

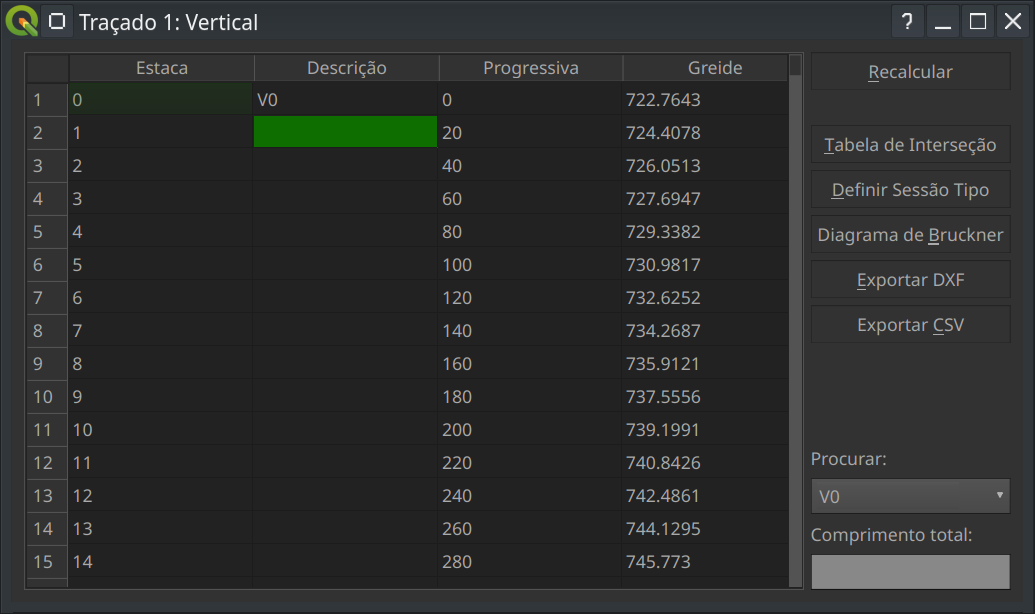
  
Figura 14: Janela com a tabela de estacas horizontais

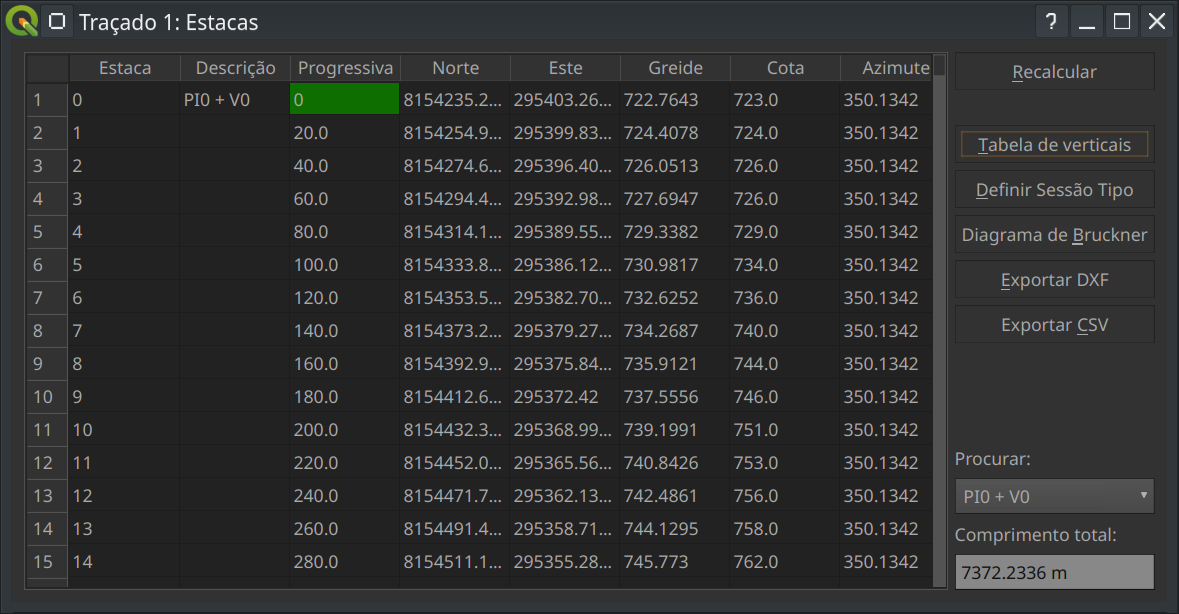
  
Figura 15: Gerenciamento de arquivos de projeto com múltiplos traçados.

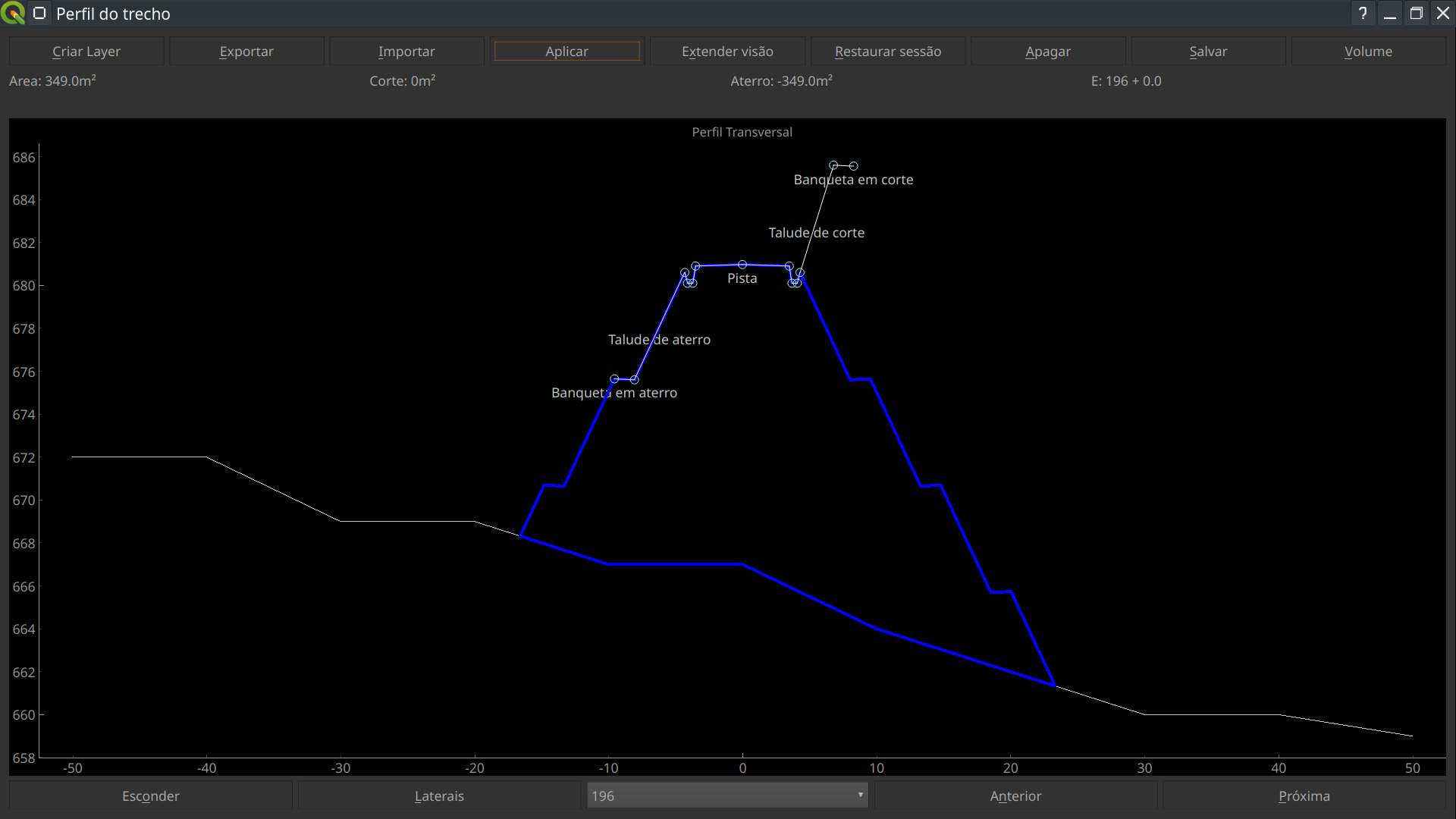
  
Figura 16: Exemplo de curva horizontal.

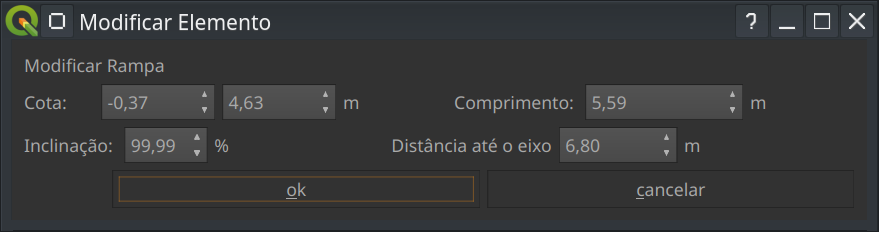
  
Figura 17: Interface para edição do greide. Os pontos azuis representam estacas horizontais no terreno, a linha em vermelho é o greide. O eixo das abscissas representa a progressiva em metros.

  
Figura 18: Definição de curvas verticais.

  
Figura 19: Tabela de estacas verticais.

  
Figura 20: Tabela de interseção apresentando a totalidade das estacas verticais e horizontais.

  
Figura 21: Interface de edição da sessão tipo.

  
Figura 22: Janela de edição de rampas usada em ambas as interfaces de perfil vertical e sessão transversal.

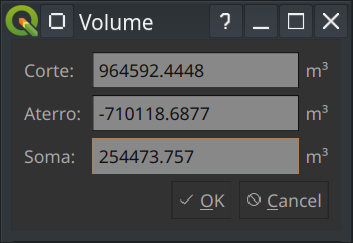
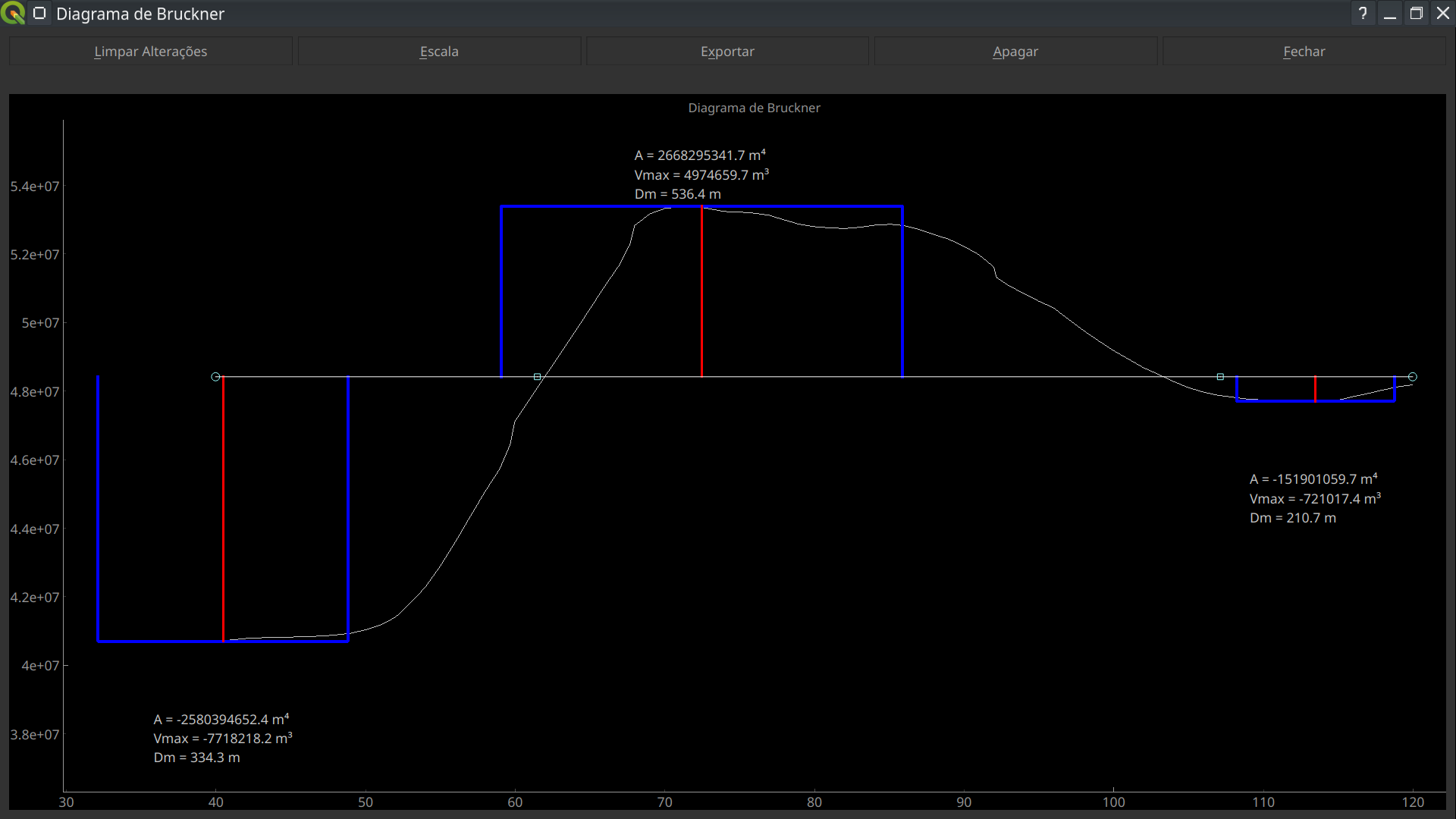
  
Figura 23: Resultados do cálculo de volume.

  
Figura 24: Diagrama de Bruckner para todo o trecho. As abscissas são as estacas e as ordenadas o volume em m³.

  
Figura 25: Curva de Bruckner e distância média.