



MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Uso de linguagem de script para realizar operações de álgebra de mapas da TerraLib5

Luan Gaspar Pinto de Melo

Objetivo

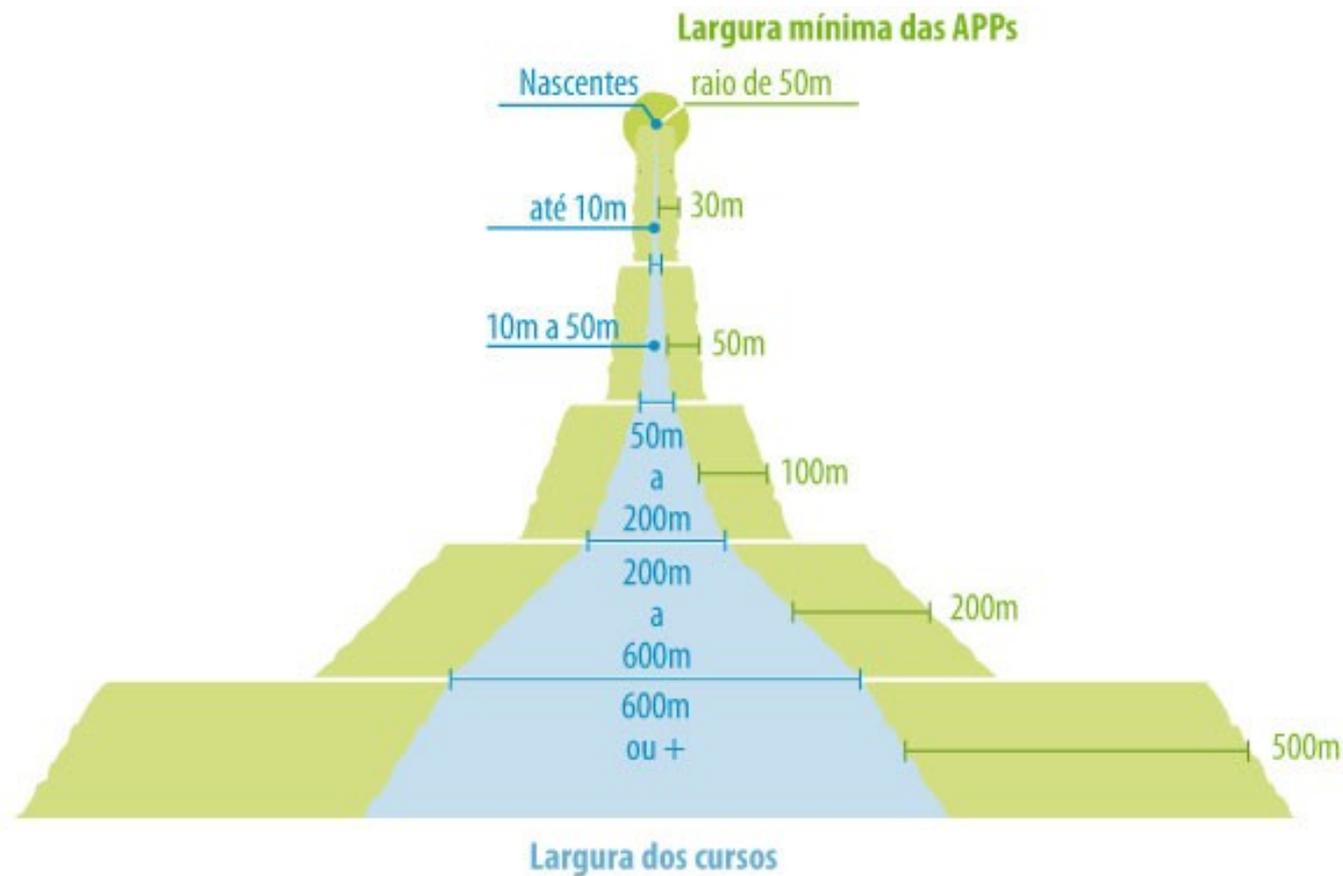
Estudar como álgebras de mapas podem se apresentar para o usuário final e discutir quais as implicações das diferentes maneiras existentes.

Porque álgebra de mapas?

- * Operações fundamentais
- * Gerar novas informações
- * Exemplo implementado Buffer

Exemplo de aplicação de Buffer

Lei nº12.651/12



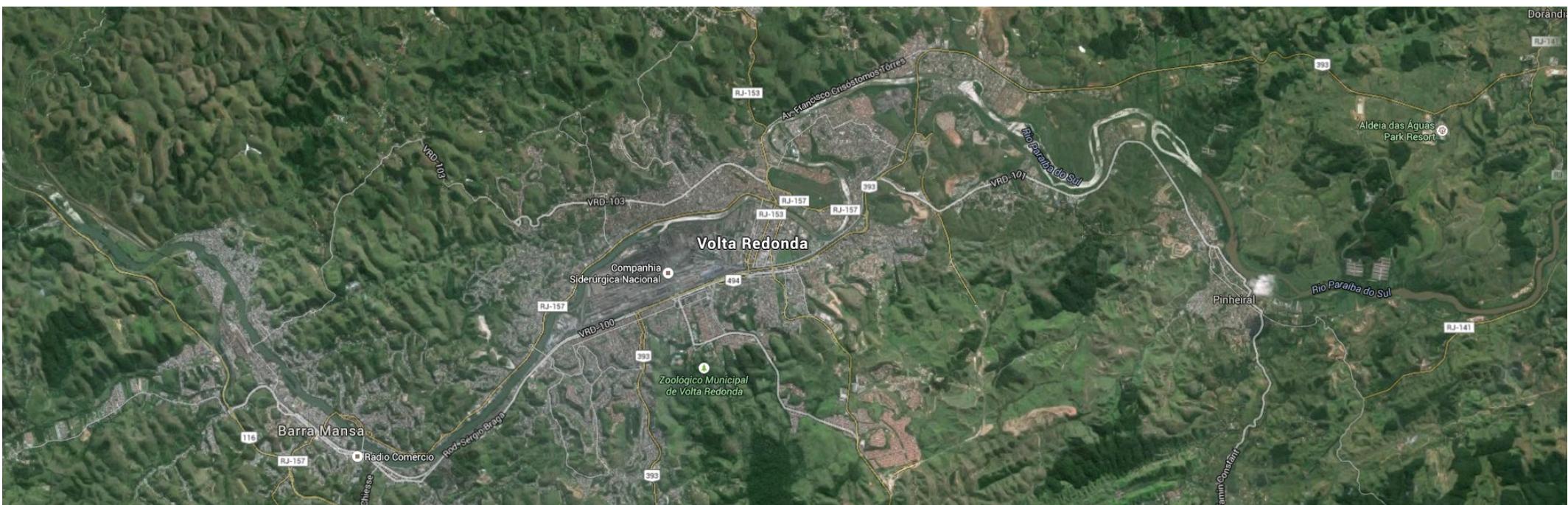
Referência: <http://www.senado.gov.br/noticias/Jornal/emdiscussao/codigo-florestal/areas-de-preservacao-permanente.aspx>

SIGs (Sistema de Informações Geográficas)



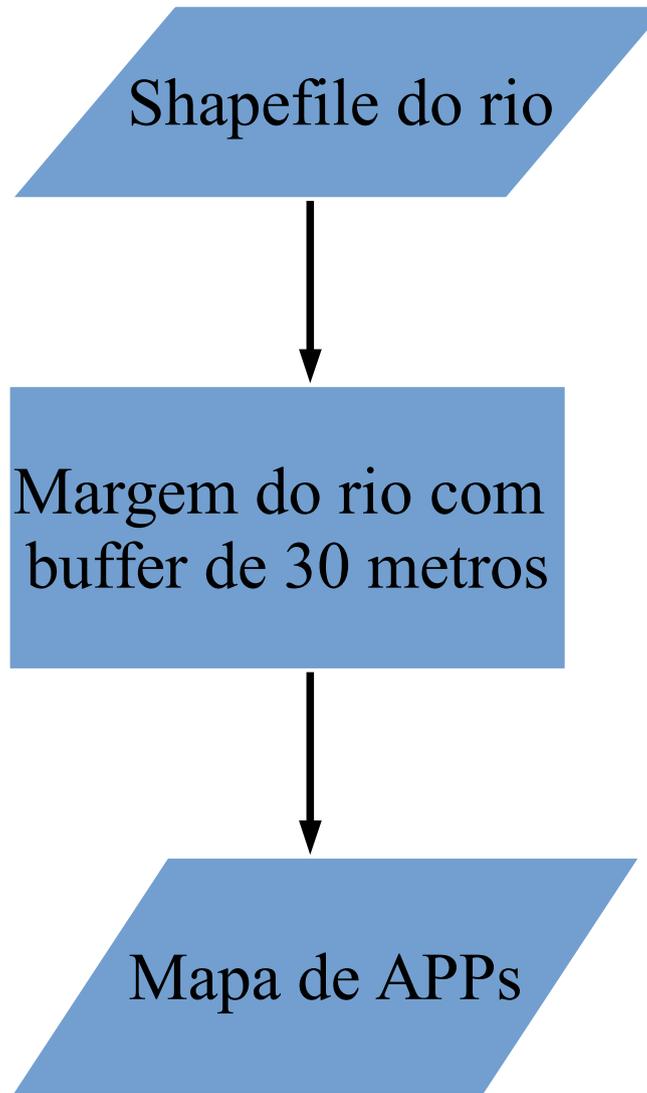
Exemplo para aplicação

Rio Paraíba do Sul no Município de Volta Redonda - RJ



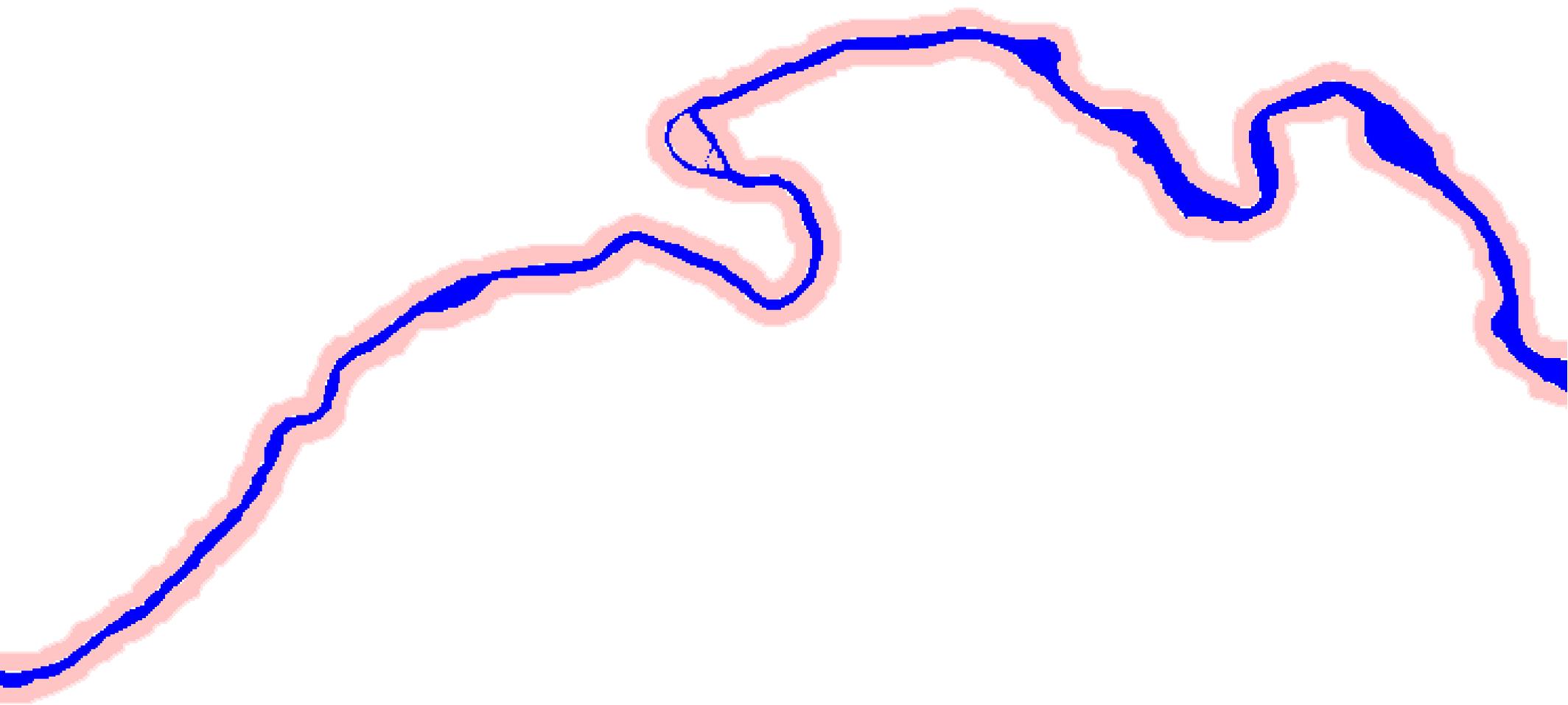
Fonte: Google Earth

Exemplificação

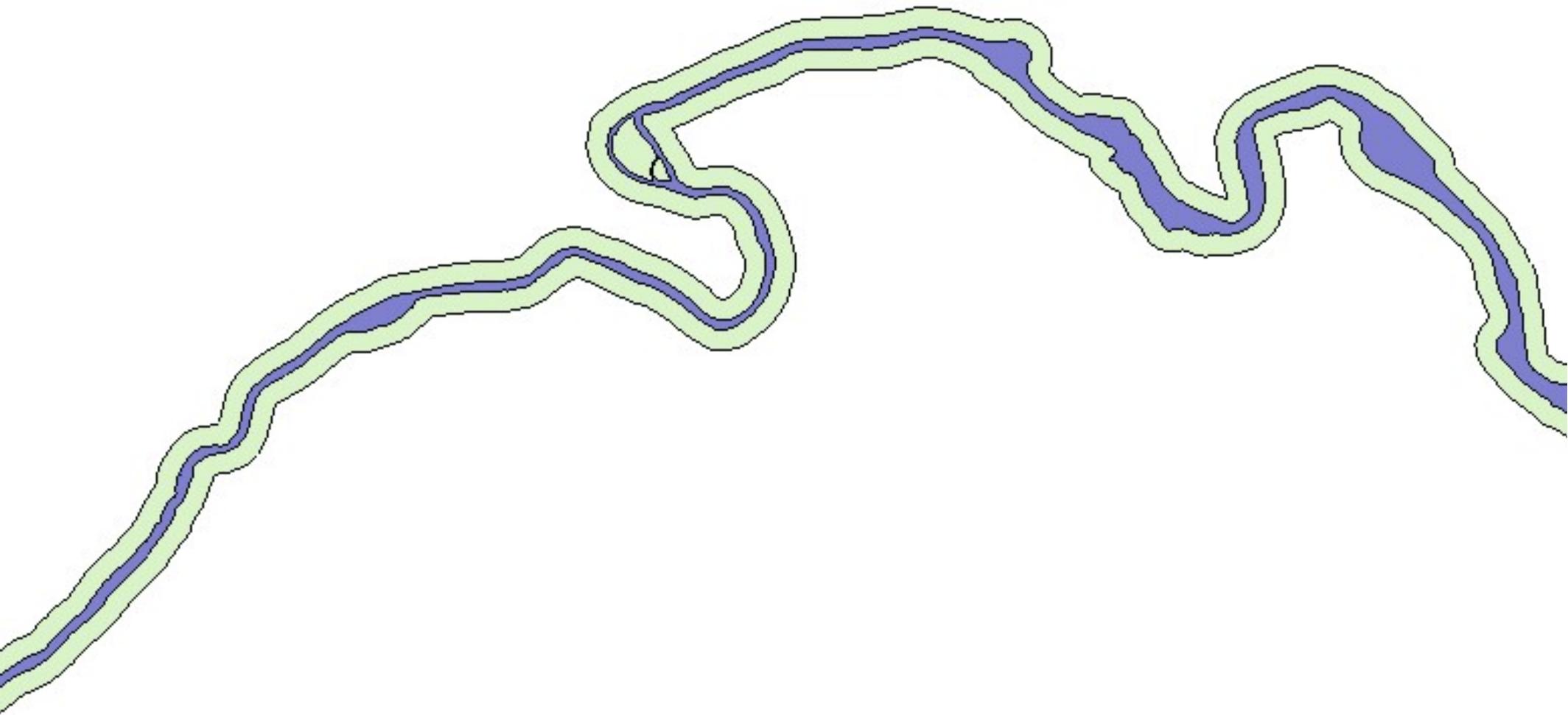


Fluxograma para obter mapa de APPs

Buffer gerado a partir do Spring



Buffer gerado a partir do TerraView



```

bool BufferOGRToOGR() {
    std::string data_dir = "/home/luan/Documents/Geoprocessamento/rio_paraiba_sul";
    std::string filename = data_dir + "rio_paraiba_sul.shp";

    std::map<std::string, std::string> srcInfo;
    srcInfo["URI"] = filename;
    srcInfo["DRIVER"] = "ESRI Shapefile";

    te::da::DataSourcePtr srcDs(te::da::DataSourceFactory::make("OGR"));
    srcDs->setConnectionInfo(srcInfo);
    srcDs->open();

    std::string inDsetName = "rio_paraiba_sul";
    if (!srcDs->datasetExists(inDsetName)) {
        std::cout << "Input dataset not found: " << inDsetName << std::endl;
        return false;
    }
    std::auto_ptr<te::da::DataSet> inDset = srcDs->getDataSet(inDsetName);
    std::auto_ptr<te::da::DataSetType> inDsetType = srcDs->getDataSetType(inDsetName);
    double distance = 0.1;
    int bufferPolygonRule = te::vp::ONLY_OUTSIDE;
    int bufferBoundariesRule = te::vp::DISSOLVE;
    bool copyInputColumns = false;
    int levels = 1;

    std::string file_result = data_dir + "file_result_rio_paraiba_sul.shp";
    std::map<std::string, std::string> tgrInfo;
    tgrInfo["URI"] = file_result;
    tgrInfo["DRIVER"] = "ESRI Shapefile";

    te::da::DataSourcePtr trgDs(te::da::DataSourceFactory::make("OGR"));
    trgDs->setConnectionInfo(tgrInfo);
    trgDs->open();

    std::string outDS = "file_result_rio_paraiba_sul";
    if (trgDs->datasetExists(outDS)) {
        std::cout << "A dataset with the same requested output dataset name already exists: " << outDS << std::endl;
        return false;
    }

    te::vp::BufferOp* bufferOp = new te::vp::BufferMemory();
    bufferOp->setInput(srcDs, inDsetName, inDsetType);
    bufferOp->setOutput(trgDs, outDS);
    bufferOp->setParams(distance, bufferPolygonRule, bufferBoundariesRule, copyInputColumns, levels);

    bool result;
    if (!bufferOp->paramsAreValid())
        result = false;
    else
        result = bufferOp->run();
    delete bufferOp;
    return result;
}

```

O que acontece quando o usuário clica em um botão para gerar um buffer?

Proposta

* Propor uma API Python que obtenha acesso à álgebra de mapas utilizando as funcionalidades fornecidas pela TerraLib5.

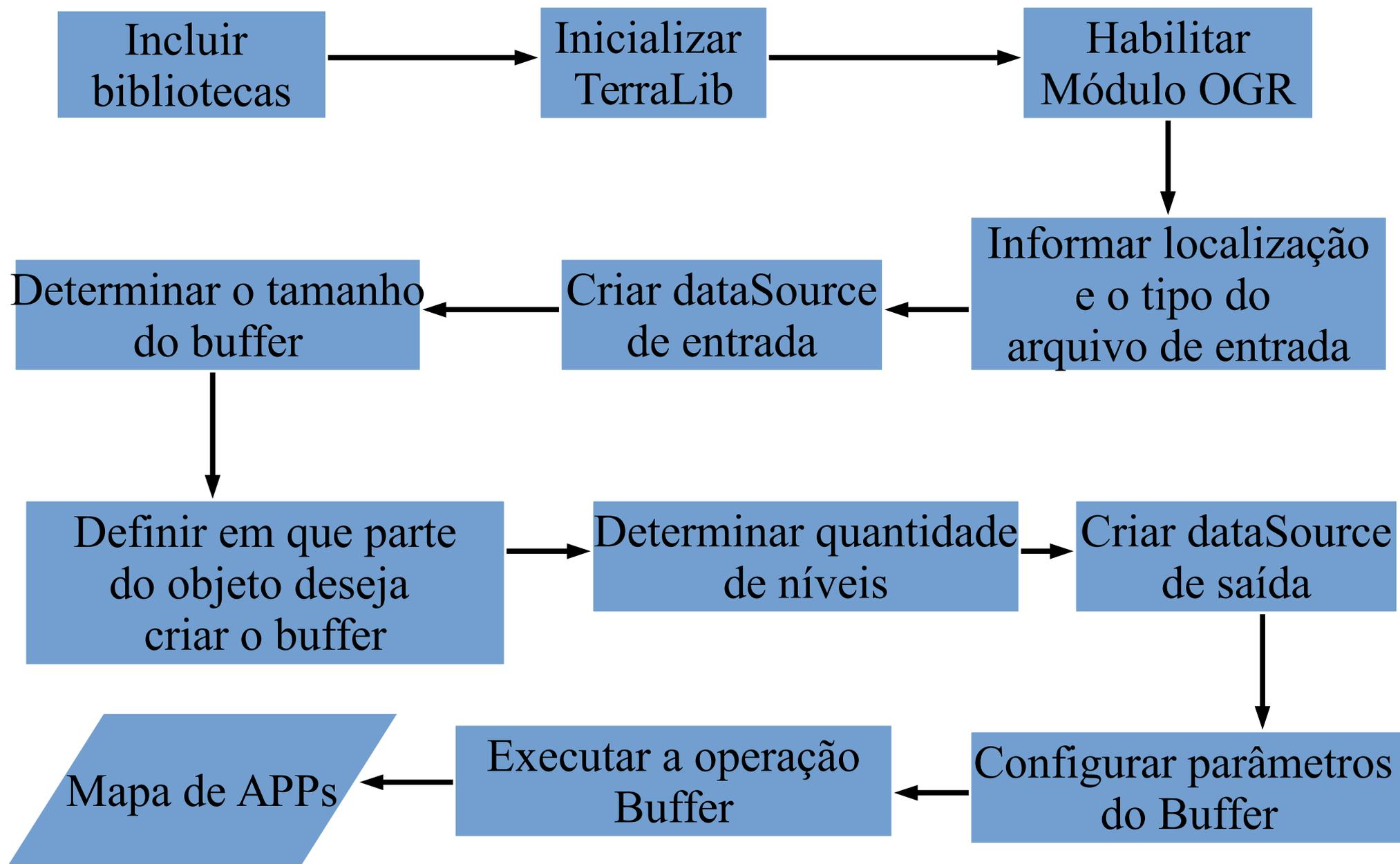
Vantagem

* Maior flexibilidade e eficiência na implementação de metodologias que combinam diferentes dados para obtenção de um resultado específico. Por exemplo, realizar várias operações de álgebra de mapas com um único script.

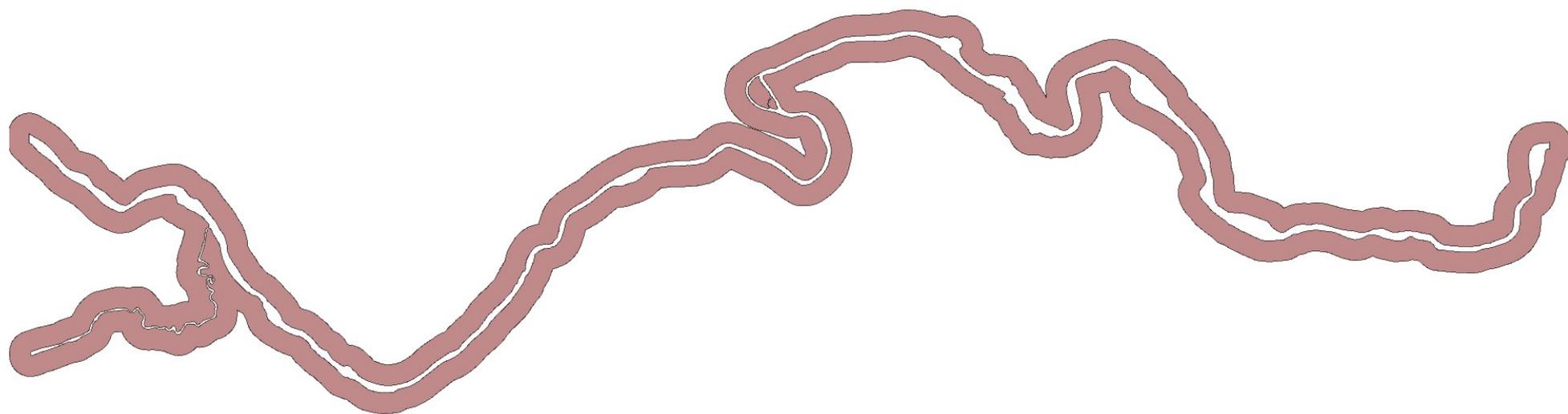
Desafios

- * Como configurar e aprender sobre a TerraLib?
 - Instalar todas as dependências (bibliotecas de terceiros)
 - Baixar a TerraLib completa
 - Compilar a TerraLib
 - Configurar a IDE para fazer o link com a biblioteca TerraLib
 - Adicionar os módulos que serão utilizados na aplicação
 - Configurar as variáveis de ambiente
 - Desenvolver o código para realizar a operação desejada

Fluxograma do código para realizar a operação Buffer



Buffer obtido através do programa em C++



O que fazer para utilizar a linguagem de script Python?

- * Analisar parâmetros de entrada
- * Adaptar o código em C++
- * Gerar o Bind

Exemplo de como seria um script em Python para executar a operação Buffer

```
Import buffer
```

```
BufferOGRTToOGR(arquivo, posicaoBuffer, regraLimite, niveis, arquivoSaida)
```

arquivo="caminho do arquivo de entrada"

posicaoBuffer= externo, interno ou internoExterno

regraLimite= "dissolver" ou "nao dissolver"

niveis = 1

arquivoSaida= "nome do arquivo de saída"

Resultados

- * Entendi o C++
- * Entendi a TerraLib
- * Consigo programar na TerraLib
- * Domínio da aplicação

Próximos passos

- * Criar Módulo em C++ para Python
- * Compilar o Módulo
- * Instalar o módulo no Python
- * Utilizar o módulo criado em um script Python