Identificador de Campos de Shapefiles

A seguir é apresentado a descrição e elaboração da ferramenta de Identificador de Campos de Shapefiles.

SUMÁRIO

[1. Introdução 3](#_Toc469387968)

[2. Objetivos 3](#_Toc469387969)

[3. Desenvolvimento da primeira versão da ferramenta 3](#_Toc469387970)

[4. Desenvolvimento da segunda versão da ferramenta 8](#_Toc469387971)

[4.1. Ajustando e executando a ferramenta 9](#_Toc469387972)

# Introdução

No presente documento será descrito como foi gerada a ferramenta “Identificador de campos de Shapefiles”, cujo objetivo é listar em um arquivo CSV os arquivos shapefiles com as informações dos seus campos como: nome, tipo e extensão.

# Objetivos

O primeiro passo foi analisar e identificar quais seriam os processos necessários afim de que a ferramenta atingisse o objetivo que era listar em um arquivo CSV os arquivos shapefiles com as informações dos seus campos como: nome, tipo e extensão.

Em função disto, foi desenvolvido um fluxograma que serviu de mapeamento dos processos a ser desenvolvidos. Como pode ser observado na Figura 1.

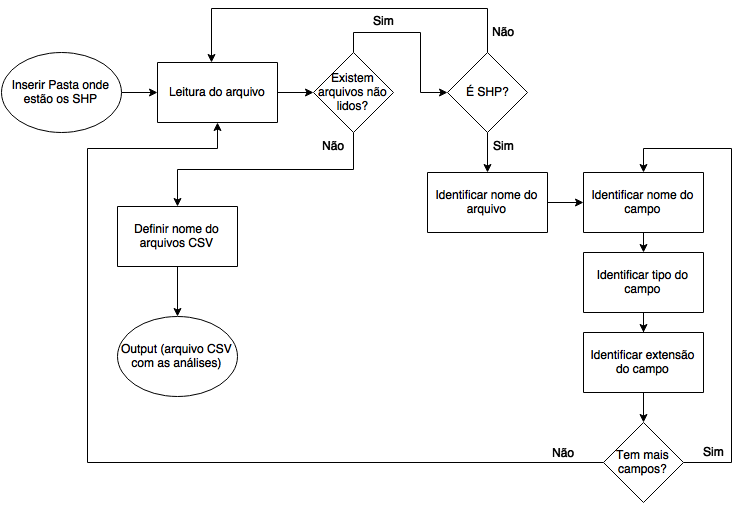


Figura : Fluxograma da ferramenta.

# Desenvolvimento da primeira versão da ferramenta

Tendo pleno conhecimento dos processos a ser seguido, pôde-se iniciar a construção do script que deu base a ferramenta. Esta ferramenta foi desenvolvida na linguagem de programação Python, por isso é composta por um arquivo com extensão “.py”.

Tabela : Algoritmo do arquivo “ListShapeAndFields.py”.

|  |
| --- |
| **import** arcpy  **import** csv  arcpy**.**env**.**workspace **=** input**(**'Inserir endereço onde estão salvos os Shapefiles: '**)**  shapes **=** arcpy**.**ListFeatureClasses**()**  **def** toASCII**(**str**):**  **return** str**.**encode**(**'ascii'**,** 'ignore'**).**decode**(**'ascii'**);**  **with** open**(**input**(**'Nome do arquivo csv: '**),** 'wb'**)** **as** csvfile**:**  streamWriter **=** csv**.**writer**(**csvfile**,** delimiter**=**','**)**  streamWriter**.**writerow**([**'Nome arquivo'**,** 'Coluna'**,** 'Tipo'**,** 'Extensão'**])**  **for** shape **in** shapes**:**  fields **=** arcpy**.**ListFields**(**shape**)**  **for** field **in** fields**:**  streamWriter**.**writerow**([**toASCII**(**shape**),** toASCII**(**field**.**name**),** field**.**type**,** field**.**length**])** |

Para executar a ferramenta, é necessário executar com o IDLE o arquivo de script gerado “ListShapeAndFields.py” e usar a ferramenta “Run Module”. Podendo usar a tecla “F5” como atalho.

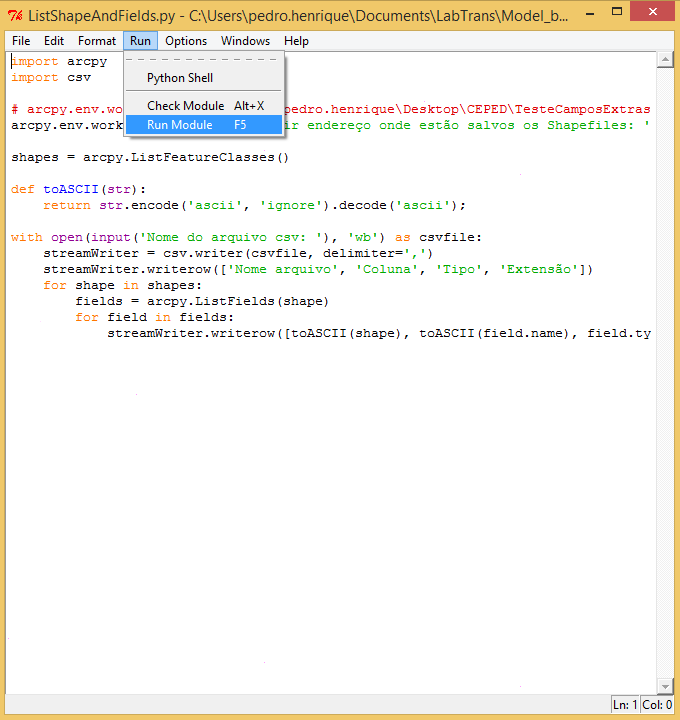


Figura : Ambiente IDLE do Python.

Em seguida apenas é necessário preencher os parâmetros solicitados pela ferramenta. Sendo o primeiro, o endereço onde estão salvos os arquivos Shapefiles. Deve ser lembrado que as respostas devem estar acompanhadas de aspas simples (‘’).

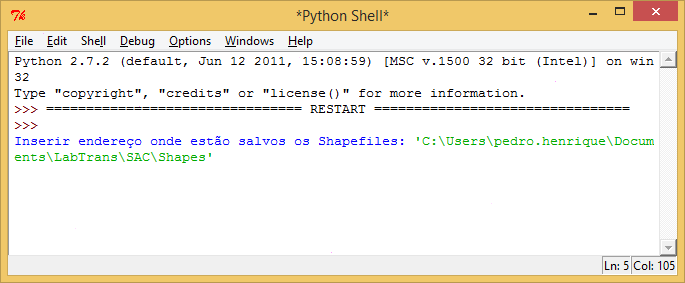


Figura : Ferramenta com o primeiro parâmetro solicitado.

Por fim, deve-se indicar o nome do arquivo CSV a ser utilizado, por exemplo: ‘analiseCampos.csv’.

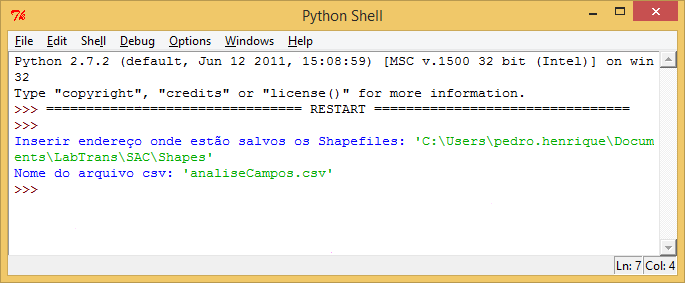


Figura : Ferramenta com o segundo parâmetro solicitado.

Ao final deste processo é gerado o arquivo CSV (*Comma-separated values*), que como o nome já diz, é um arquivo separado por vírgulas. Para que as funções de um arquivo XLSX sejam utilizadas no arquivo gerado, é sugiro a utilização da ferramenta “Texto para colunas” do software Excel.

Na primeira etapa da ferramenta deve ser escolhido “Delimitado” como o tipo de campo que melhor descreva os dados.

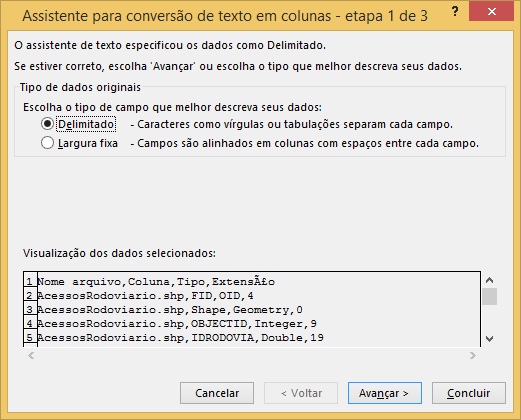


Figura : Primeira etapa da ferramenta “Texto para Colunas”.

Na segunda etapa, deve ser escolhido os símbolos delimitadores. Para este caso deve ser selecionado apenas a “Vírgula”.

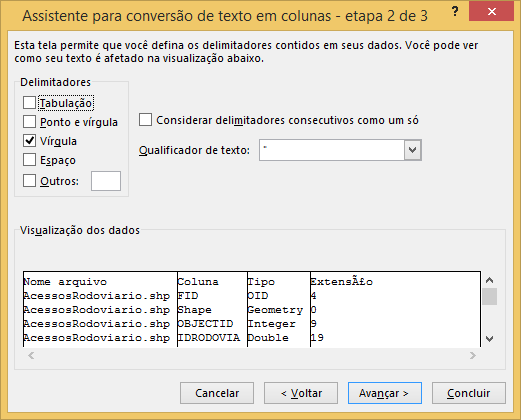


Figura : Segunda etapa da ferramenta “Texto para Colunas”.

Por fim, deve ser utilizado o formato dos dados da coluna.

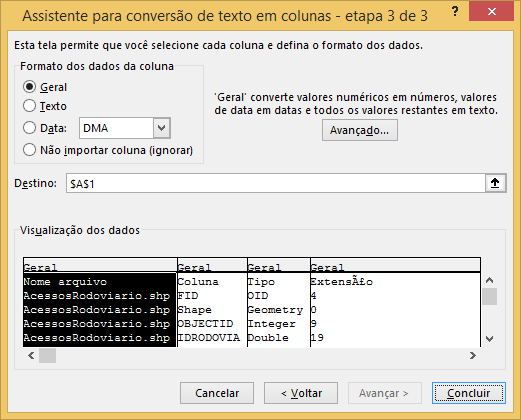
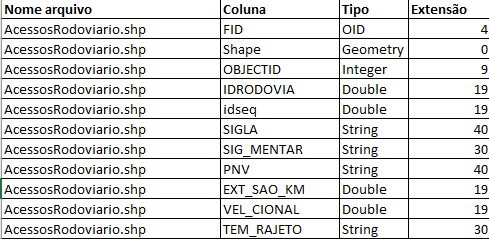


Figura : Terceira etapa da ferramenta “Texto para Colunas”.

Como resultado, é uma planilha separada por quatro colunas, sendo elas: nome do arquivo, campo, tipo e extensão.

Tabela : Resultado gerado pela ferramenta “ListShapeAndFields.py” acrescido da ferramenta “Texto para Colunas”.



# Desenvolvimento da segunda versão da ferramenta

Após a conclusão da ferramenta “ListShapeAndFields.py”, foi gerado uma segunda versão com o nome “shapes-to-csv”. Esta ferramenta também foi desenvolvida na linguagem Python e é composta por dois arquivos de mesmo nome e com extensões “.py” e “.bat” que devem estar na mesma pasta.

O arquivo “shapes-to-csv.py” é onde está o algoritmo lógico da ferramenta. Ela não é auto executável, mas deve estar sempre no mesmo endereço do arquivo “.bat”.

Tabela : Algoritmo do arquivo “shapes-to-csv.py”.

|  |
| --- |
| #!/usr/bin/env python  # -\*- coding: utf-8 -\*-  **import** os  **import** sys  **import** csv  **import** arcpy  args **=** sys**.**argv**[**1**:]**  # Functions definitions  **def** getPath**(**args**):**  **return** args**[**0**];**  **def** getFileName**(**args**):**  **return** '{0}.csv'**.**format**(**args**[**1**]);**  **def** goTo**(**path**):**  os**.**chdir**(**path**)**  **return;**  **def** toASCII**(**str**):**  **return** str**.**encode**(**'ascii'**,** 'ignore'**).**decode**(**'ascii'**);**  **def** writeCSV**(**arcpy**,** fileName**):**  **with** open**(**fileName**,** 'wb'**)** **as** csvfile**:**  streamWriter **=** csv**.**writer**(**csvfile**,** delimiter**=**','**)**  streamWriter**.**writerow**([**'Nome arquivo'**,** 'Coluna'**,** 'Tipo'**,** 'Extensão'**])**  shapes **=** arcpy**.**ListFeatureClasses**()**  **for** shape **in** shapes**:**  fields **=** arcpy**.**ListFields**(**shape**)**  **for** field **in** fields**:**  streamWriter**.**writerow**([**toASCII**(**shape**),** toASCII**(**field**.**name**),** field**.**type**,** field**.**length**])**  **return;**  # Main  arcpy**.**env**.**workspace **=** getPath**(**args**)**  goTo**(**getPath**(**args**))**  writeCSV**(**arcpy**,** getFileName**(**args**))** |

A grande vantagem desta segunda edição é a proteção do código, visto que para executá-la não é necessário abrir a IDLE e sim apenas executar via CMD (*Command Prompt*) do Windows. Esta execução ocorre através do arquivo “shapes-to-csv.bat”.

Tabela : Algoritmo do arquivo “shapes-to-csv.bat”.

|  |
| --- |
| @**echo** off  **set** /p path**=**"Digite o caminho para os arquivos Shapefiles -**>** "  **set** /p name**=**"Digite o nome para o arquivo que será gerado -**>** "  shapes-to-csv.py **%path%** **%name%**  **pause** |

# Ajustando e executando a ferramenta

O primeiro uso da ferramenta exige a configuração do Python como “*path*” do Windows. Para executar essa alteração deve ser acessado as configurações do sistema através do atalho “Windows + Pause”. E em seguida acessar o campo “Configurações avançadas do sistema”.

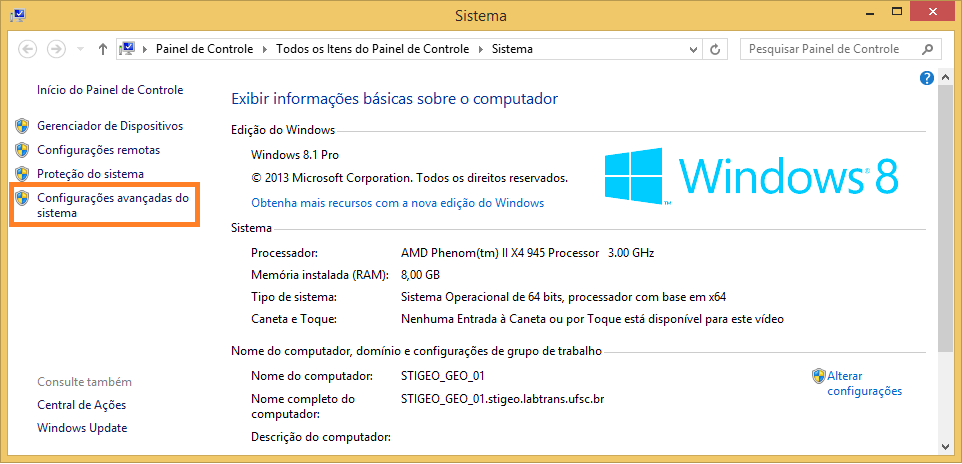


Figura : Tela de Configurações do Sistema.

O segundo passo é clicar em “Avançado” e em seguida em “Variáveis de Ambiente”.

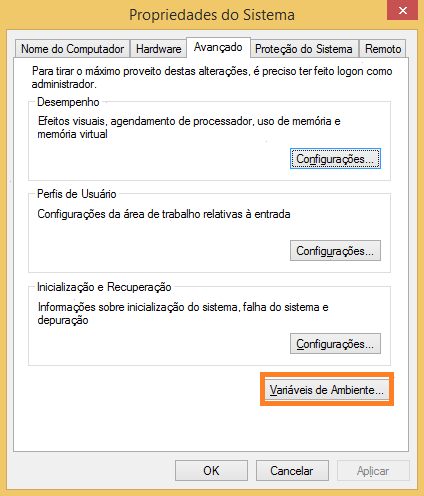


Figura : Tela de Propriedades do Sistema.

O terceiro passo é em “Variáveis do sistema” buscar a variável “Path”.

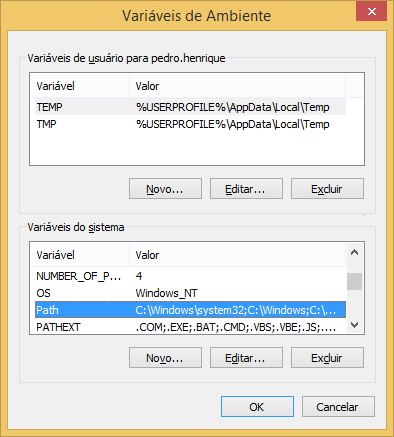


Figura : Tela de Variáveis do Ambiente.

O quarto e último passo é clicar em editar e ao final do valor apresentado, acrescentar “**;C:\Python27\ArcGIS10.1\”.**

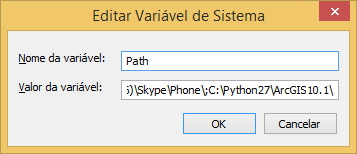


Figura : Tela de edição de variável de sistema.

Feita a configuração do “path” do Python no Windows a ferramenta fica apta a uso. Para a correta utilização da ferramenta deve-se primeiramente dar dois cliques no arquivo “shapes-to-csv.bat”.

Em seguida, é aberta a janela “cmd.exe” com o primeiro requisito da ferramenta: digitar o endereço onde constam os arquivos Shapefiles. Nesta versão não é necessário o uso de aspas.

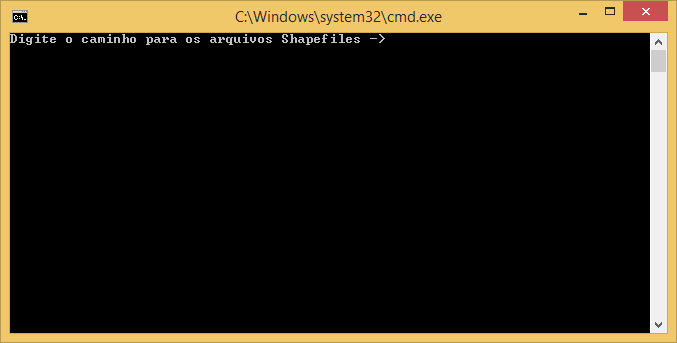


Figura : Primeiro parâmetro da Ferramenta “shapes-to-csv”.

Em seguida é apresentado o segundo argumento: “Digite o nome para o arquivo que será gerado.

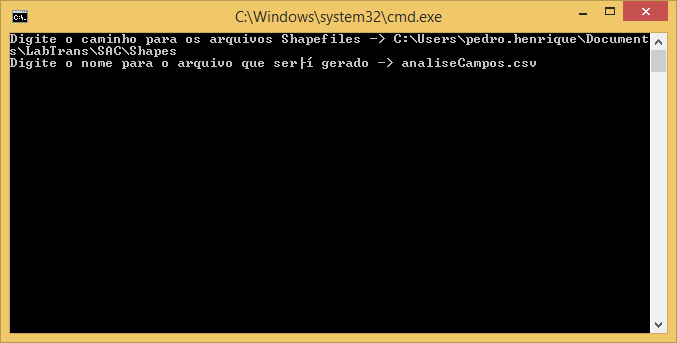


Figura : Segundo parâmetro da Ferramenta “shapes-to-csv”.

Por fim é necessário apenas pressionar qualquer tecla para continuar.

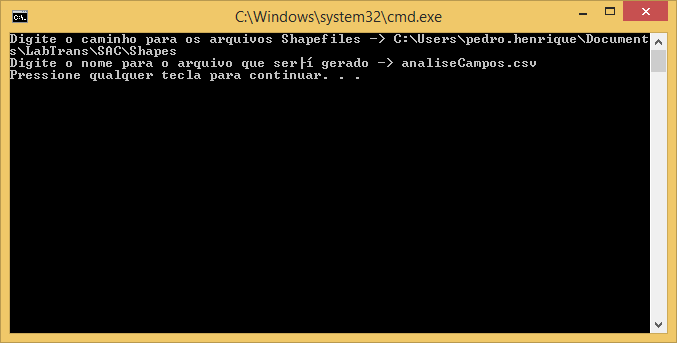


Figura : Finalização da ferramenta.

Ao final deste processo, assim como para a primeira edição pode-se ter o arquivo CSV com os campos: nome do arquivo, campo, tipo e extensão.

Tabela : Resultado gerado pela ferramenta “ListShapeAndFields.py” acrescido da ferramenta “Texto para Colunas”.

