

## AzimuthFinder: ferramenta para a extração de dados e apoio na análise estrutural

*AzimuthFinder: a software for data reading in structural geology*

Gustavo Lopes Queiroz<sup>1</sup>, Eduardo Salamuni<sup>1</sup>, Ednilson Roberto do Nascimento<sup>1</sup>

<sup>1</sup>Grupo de Neotectônica, Departamento de Geologia, Universidade Federal do Paraná - UFPR, Centro Politécnico - Jardim das Américas, Caixa Postal 19001, CEP 81531-990, Curitiba, PR, BR (gustavo.lopes.queiroz@gmail.com; salamuni@ufpr.br; deni\_ern@ufpr.br)

Recebido em 06 de maio de 2013; aceito em 27 de novembro de 2013

### Resumo

Apresenta-se uma ferramenta computacional denominada de AzimuthFinder, para funcionamento dentro do conjunto de programas do ArcGis®. O programa desenvolvido na linguagem *Python* tem por finalidade facilitar e otimizar a geração de tabelas de dados azimutais, a partir de lineamentos estruturais previamente traçados em imagens de satélite, fotografias aéreas ou mapas geológicos em meio digital. As características do *AzimuthFinder* permitem que, a critério do usuário, sejam geradas diferentes tabelas para o mesmo objeto de análise, dependendo do peso relativo atribuído aos lineamentos, por exemplo dando maior peso àqueles de traçado mais longo. Outra característica da ferramenta é a exportação dos dados no formato TXT, que é universal para vários tipos de programas, com formatação específica para *softwares* como o Stereo32, *Win Tensor*, *OpenStereo* e *StereoNet 7*, que por sua vez são *softwares* livres preparados para a confecção de diagramas de rosetas. Os testes com a ferramenta demonstraram que é bastante eficiente e rápida para a geração das tabelas azimutais, facilitando de maneira eficaz a confecção de diagramas de rosetas necessários à análise estrutural em áreas essencialmente submetidas à deformação frágil.

**Palavras-chave:** *Software*; Análise da deformação; Lineamentos estruturais; Tabelas azimutais; Digrama de rosetas.

### Abstract

Here is presented a computational tool named AzimuthFinder, for functioning inside the set of programs of ArcGis®. The developed program is intended to facilitate and optimize the generation of azimuth data tables, using structural lineaments previously traced in maps that are being worked on that automatic information system. The characteristics of AzimuthFinder allow that, upon the user's choice, different tables get generated for the same analysis object, depending on the relative weight attributed to the lineaments, giving for example greater weight to those of greater extension. Another characteristic of the tool is the file exportation in TXT format, which is universal to several types of programs, with specific formatting to one the software between Stereo32, Win Tensor, OpenStereo and StereoNet7, which are all free software prepared to the confection of rose diagrams. Tests with the program showed that it is very efficient and fast for generating the azimuth data tables, effectively allowing easier confection of rose diagrams, which are necessary for the structural analysis in areas submitted to fragile deformation.

**Keywords:** Software; Deformation analysis; Structural lineaments; Azimuth tables; Rose diagram.

## INTRODUÇÃO

Desde o advento da obtenção de fotografias aéreas verticais entre as décadas de 1940 e 1950, a ciência geológica vem utilizando-as como ferramentas para o mapeamento de domínios lito-estruturais e/ou de lineamentos estruturais. O método de foto-análise no Brasil tomou um importante impulso a partir de sua sistematização por Soares e Fiori (1976), e até hoje é base para a determinação de feições morfo-estruturais. O traçado de estruturas lineares é um dos métodos que têm sido empregados, para a determinação do arcabouço estrutural, principalmente em áreas afetadas por tectônica frágil.

Importantes trabalhos relativos ao método de traçado, sua aplicação e suas limitações aparecem em Haman (1961), Huntington (1969), Soares e Fiori (1976), Paradella e Vitorello (1981), Gama (1982), Santos (1986), Liu (1984, 1987), Ricomini e Crôsta (1988), Taylor (1988), Gillespie et al. (1993), Yamagushi (1985), Strieder e Amaro (1997), Guo, George e Lindsey (1999), Justo (2006) e Fonseca e Corrêa (2011). Alguns desses trabalhos, principalmente aqueles mais recentes, estão focados na discussão do emprego de ferramentas computacionais, visto que as imagens deixaram de ser disponibilizadas no formato impresso (ou analógico) em favor das imagens digitais (ou eletrônicas).

A partir de meados da década de 1970, principalmente nas décadas de 1980 e 1990, a utilização de sensoriamento remoto ampliou as possibilidades de observações, principalmente em relação à resolução espacial (tamanho do *pixel*) e à abrangência da área analisada. Adicionalmente, com o rápido desenvolvimento da informática, houve um avanço exponencial no processamento de imagens digitais, cujo limite se dá na obtenção da quantidade de informação por unidade de *pixel*. Ou seja, quanto melhor o equipamento utilizado para o processamento e imageamento, maior a rapidez e o nível de detalhe da análise.

Ao longo desse período, os vários *softwares* de geoprocessamento e/ou processamento de imagem melhoraram e ampliaram suas ferramentas computacionais, com o objetivo de aproveitar ao máximo a qualidade das imagens disponíveis, possibilitando inclusive, que o usuário desenvolva e utilize ferramentas personalizadas.

Assim, atualmente se tornou muito menos moroso trabalhar com imagens em multiescalas quando se objetiva traçar as feições geológicas em fotografias aéreas, imagens de satélite e até em modelos tridimensionais da superfície obtidos por radar ou estereoscopia óptica. Outro fato relevante em relação ao avanço tecnológico na obtenção de dados da superfície terrestre por sensores remotos, diz respeito à cobertura de praticamente toda a superfície terrestre por levantamentos topográficos e imagens de diferentes

resoluções espaciais, temporais, espectrais e radiométricas. O que permitiu identificar e analisar deformações crustais em qualquer ponto do planeta.

Todavia, se os *softwares* atuais são excelentes ferramentas para a manipulação das imagens no que tange à mudança de cor, na possibilidade em trabalhar com múltiplas camadas (*layers*), na possibilidade da rápida alternância de escala — planar ou tridimensional — e no oferecimento de ferramentas que possibilitam o traçado de elementos que definem a analogia e a homologia na foto-análise, os programas são limitados em relação a algumas necessidades específicas à análise estatística de dados geológico-estruturais, obtidos em ambiente de Sistema de Informação Geográfica (SIG).

Um exemplo clássico é a dificuldade da seleção automática e individualização de áreas poligonais irregulares, ou zonas homólogas, que requerem um aparato mínimo de inteligência artificial bem como a utilização de mais de um *software*, principalmente quando se trabalha com Sistemas de Informação Geográfica (Bettú, 2012). Outro exemplo limítrofe, porém menos complexo, refere-se à construção de diagramas estruturais simples, tais como o Diagrama de Rosetas (ou rosáceas) que tem por finalidade mostrar estatística direcional de lineamentos e/ou lineamentos retilíneos ou pouco curvilíneos.

É comum, por exemplo, o geólogo estruturalista ou o geomorfólogo estrutural, após ter realizado cuidadoso trabalho de foto-análise, voltado aos aspectos geológico-estruturais, ter que recorrer a procedimentos pouco eficientes e demorados para a quantificação de dados digitais. Há dificuldade prática no procedimento de montar uma tabela numérica de dados estruturais azimutais com o intuito de utilizá-la em *softwares* específicos de análise estrutural para a confecção de Diagramas de Rosetas.

Tal dificuldade implica na morosidade do trabalho de interpretação, posto que os pesquisadores tem sido levados a exportar os dados digitais para outros *softwares* do tipo CAD, não plenamente amigáveis com os programas de geoprocessamento. Isso resulta no uso de ferramentas simplistas para obtenção dos dados direcionais, ou então na impressão de informações gráficas para, em processo artesanal, medir manualmente o ângulo que os traços dos lineamentos fazem com a referência geográfica (em geral o Norte verdadeiro).

Além disso, uma dificuldade adicional é que mesmo que se quantifiquem de forma automática os lineamentos traçados, o resultado não prevê a atribuição de pesos de acordo com a importância geológico-estrutural dos lineamentos melhor marcados e/ou mais longos. Por exemplo, um lineamento de 1 km terá o mesmo peso (será marcado como uma unidade) que um lineamento de 10 km. Adicionalmente, as ferramentas automáticas de extração de dados quantificados, em

geral, não possuem como característica intrínseca levar em consideração a mudança de ângulo do lineamento, ou seja, toma como base o ponto final e o ponto inicial do segmento traçado e considera que a direção é representada por uma reta simples entre os dois pontos. Isto leva a um erro de interpretação de lineamentos que estejam flexionados devido à natureza da estrutura.

Buscando a solução desses problemas, desenvolveu-se uma ferramenta computacional, que pode ser considerada como um *script* agregado a outro de maior complexidade. O objetivo é medir espacialmente os lineamentos traçados em imagens, em relação ao norte verdadeiro e, conseqüentemente, montar automaticamente as tabelas de dados azimutais necessárias à confecção de Diagramas de Rosetas.

O programa foi desenvolvido para funcionar como uma ferramenta incorporada ao conjunto de programas do ArcGis® (ESRI, 2012), recebeu a denominação de *AzimuthFinder*, e é aberto à comunidade para sua utilização, desde que citada sua fonte (disponível em: [www.neotectonica.ufpr.br](http://www.neotectonica.ufpr.br)). Por outro lado, como a programação em linguagem *Python* independe propriamente do sistema ArcGis®, a adaptação do *AzimuthFinder* a outros *softwares* de geoprocessamento poderá ser realizada de forma relativamente fácil, desde que o mesmo seja compatível com a linguagem citada.

Assim sendo, o trabalho de confecção do programa ora apresentado dentro do ArcGis®, é meramente exemplar, e aproveita algumas facilidades internas daquele *software*. Há que se considerar que o conhecimento de linguagem *Python* foi necessário, tanto quanto o entendimento do comportamento estrutural frágil de maciços e as heterogeneidades reológicas que condicionam seu maior ou menor grau de cisalhamento rúptil.

## OBJETIVO E JUSTIFICATIVA

O trabalho tem por objetivo apresentar o desenvolvimento do *software* denominado *AzimuthFinder*, programado em *Python*, desenhado e adaptado a título de teste de aplicação para integrar-se como ferramenta ao conjunto de programas ArcGis®. Seu desenvolvimento é justificado pela necessidade de uma ferramenta computacional eficiente e rápida para a quantificação de dados de lineamentos estruturais, atentando para suas características inerentes como comprimento, frequência, inflexões e orientações preferenciais.

Adicionalmente, pretendeu-se com o presente trabalho, tornar a ferramenta pública (Anexos 1 e 2), para que outros usuários e pesquisadores a utilizem de forma aberta, inclusive desenvolvendo-o para maior sofisticação que desejarem.

## MATERIAIS E MÉTODOS

Para a programação do *software* foi utilizado o ambiente de desenvolvimento integrado para *Python* (IDLE) integrado ao ArcGis®, e para testar o programa foram utilizados diversos arquivos de *shape*, produzidos no ArcMap. Para criar e testar os arquivos de saída foram utilizados os *softwares* Stereo32 (Roller e Trepmann, 2003), *Win Tensor* (Delvaux, 2013), *OpenStereo* (Grohmann e Campanha, 2010) e *Stereonet 7* (Allmendinger, 2011), voltados à geologia estrutural.

Como referência para a programação em *Python*, foram realizadas consultas ao site oficial da linguagem <<http://www.Python.org/doc/>> e como referência para nomenclaturas e estruturas de dados específicas do ArcGIS® foi consultado a página eletrônica <<http://help.arcgis.com/en/arcgisdesktop/10.0/help/>>.

## DESENVOLVIMENTO

Já existem códigos que são disponibilizados na rede da *internet*, para uso associado ao ArcMap (como funções do *Field Calculator*), os quais calculam os azimutes de linhas traçadas em um mapa. Porém, são códigos limitados, que não levam em consideração o comprimento das linhas e não geram arquivos prontos para a produção de diagramas, além de ser pouco amigáveis para os usuários com pouco conhecimento em edição de dados espaciais.

Com o intuito de desenvolver o *software* *AzimuthFinder*, que no caso do ArcMap, constitui ferramenta integrada na *toolbox*, foi utilizado o *Python* (IDLE) para facilitar o trabalho da produção de Diagramas de Rosetas. É importante salientar que o *software*, ora apresentado, não possui a capacidade de desenhar o diagrama propriamente dito, mas sim a de converter o dado gráfico em dados numéricos azimutais segundo quadrantes, ou seja, dados numéricos de direção referidos à posição Norte como considerado em análises geológicas, portanto com atribuição de 0 grau, no caso do Norte verdadeiro. Considera-se que os segmentos de linhas traçadas em mapa, no presente caso, são lineamentos estruturais de acordo com a definição de O'Leary, Freedman e Pohn (1976). A conversão por meio da ferramenta proposta gera um arquivo de texto que pode ser exportado para arquivos adaptados a diferentes *softwares* de confecção de diagramas, os quais podem ser escolhidos por dispositivo dentro do próprio *AzimuthFinder*, e que têm por objetivo a representação gráfica na forma de Diagrama de Rosetas. Os arquivos de texto gerados são abertos para edição e podem ser editados para a leitura em diversos outros *softwares* de geologia estrutural que aceitem arquivos no formato TXT.

O desenvolvimento propriamente dito seguiu as seguintes etapas: produção do código para o cálculo de azimutes, análise de *softwares* de processamento de dados estruturais e programação dos arquivos de saída, produção da interface de usuário (Figura 1) e, por fim, a realização de testes, ajustes e correções.

O cálculo do azimute das linhas é feito utilizando-se o conceito da função trigonométrica de tangente. Inicialmente o programa armazena as coordenadas X e Y do ponto inicial e do ponto final da reta. Em seguida são calculados os tamanhos dos catetos ( $X_2 - X_1$  e  $Y_2 - Y_1$ ). Em seguida o programa calcula o azimute segundo as seguintes fórmulas:

$$\text{Azimute} = 90 - \arctg(\text{Cateto Y} / \text{Cateto X}) \quad (1)$$

$$\text{Azimute} = 270 + \arctg(\text{Cateto Y} / \text{Cateto X}) \quad (2)$$

Fórmulas (1) para retas posicionadas no primeiro e terceiro quadrantes; e (2) para retas posicionadas no segundo e quarto quadrantes.

O método convencional de medidas é denominado de “frequência”, no qual para cada lineamento quando se roda o programa, será escrita uma linha com sua medida de azimute no arquivo de saída, ou seja, todos os lineamentos terão o mesmo peso, independentemente de seus tamanhos relativos.

Foi implementado um método para que os lineamentos de maior comprimento tenham maior peso que os menores, denominado método do “comprimento”, caso seja do interesse do usuário. Neste caso, o programa calcula o peso de cada lineamento. O peso é definido pelo número

de vezes aproximado em que o comprimento do lineamento é maior que o menor valor de comprimento de lineamento existente (este valor pode ser alterado arbitrariamente pelo usuário), cujo peso passa a valer 1. Sendo assim, quando o peso de um lineamento for maior que 1, o programa irá copiar seus dados (valor de azimute e mergulho), no arquivo de saída, uma quantidade de vezes igual ao seu peso. Os lineamentos cujos pesos forem menor ou igual a 1, têm seus dados copiados uma única vez no arquivo de saída. Desta forma os programas de confecção de Diagramas de Rosetas irão interpretar os valores repetidos como grandes populações de dados azimutais, o que causará uma ênfase nas direções dos lineamentos de maior peso.

Com o intuito de testar a aplicação do *Azimuth Finder*, os arquivos, gerados a partir da solicitação da ferramenta, foram preparados para programas automáticos independentes, que têm por objetivo o processamento de dados estruturais e a geração de Diagramas de Rosetas, entre outras saídas gráficas. Assim, nessa versão inicial, o arquivo de saída será apto para a utilização em um dos seguintes *softwares*: *Stereo32*, *Win Tensor*, *OpenStereo* e *Stereonet 7*, os quais foram estudados a fim de se entender qual é a estrutura específica dos arquivos de entrada de dados e, portanto, como o formato deveria ser introduzido no código do *AzimuthFinder*. É possível, no entanto, mediante ajustes no *AzimuthFinder*, prepará-lo para a saída de outros *softwares*.

Após a programação foram realizados testes em diversas máquinas com diferentes licenças do ArcGIS®, e também com mapas estruturais diferenciados. Entre os testes, foram feitos ajustes e correções finais na estrutura do *software*, estando o mesmo pronto para ser utilizado, via importação do arquivo para dentro da *toolbox*, ou pela compilação do código mostrado no Anexo 2.

## RESULTADOS E DISCUSSÃO

O estabelecimento de ferramenta computacional para determinação das direções dos lineamentos estruturais mostrou-se bastante útil no estudo de caso realizado em área interpretada entre o Sudeste do Paraná e o Nordeste de Santa Catarina (Figura 2).

Na área piloto inicialmente realizou-se o traçado de lineamentos morfo-estruturais, que no presente caso coincidem com vales e/ou segmentos de drenagem alinhados, ou seja, alinhamentos de relevo que indicam a possível presença de estruturas tectônicas rúpteis, ou seja, juntas e falhas. Com o objetivo de determinar quais as principais direções de fraquezas estruturais, considerando também os pequenos segmentos de alinhamentos

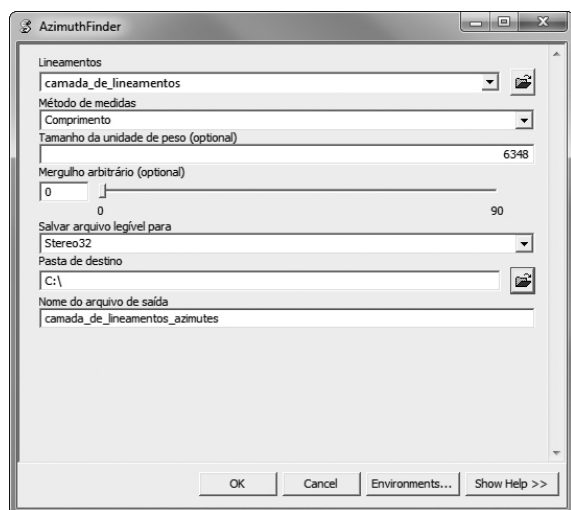


Figura 1. Interface de usuário da ferramenta AzimuthFinder.

de relevo para a interpretação, o resultado foi uma figura de certa complexidade. A profusão de traços devido à grande densidade de lineamentos e alinhamentos de relevo, ao criar dificuldades para a determinação visual de quais direções de estruturas são de fato mais representativas, motivou o desenvolvimento de uma ferramenta capaz de extrair e manipular as informações passíveis de representação gráfica.

Assim, torna-se bastante útil a análise dessas estruturas por meio do Diagrama de Rosetas que, se não tem recursos para interpretações estruturais mais avançadas, é um bom indicador inicial da deformação rúptil do maciço como resultado de um ou mais ciclos de deformações.

Todavia, como o gráfico do Diagrama de Rosetas por si só é limitado, a análise de estruturas rúpteis em maciços de diferentes reologias, apenas utilizando esse método, pode levar a dúvidas sobre seu resultado. Dessa maneira, com o intuito de melhor testar a ferramenta, após ter-se procedido ao traçado, buscou-se delimitar uma área cuja unidade geológica, pelo menos em tese, se comportasse com homogeneidade reológica, inclusive definindo uma zona homóloga na imagem. A definição de zonas homólogas ou de uma unidade geológica de interesse não é realizada automaticamente no *AzimuthFinder*, sendo necessária sua definição a partir de *softwares* específicos de geoprocessamento.

Ao importar o *AzimuthFinder* para a *toolbox* no ArcGIS®, o programa estará pronto para o uso e, quando solicitado, responderá gerando a tela de interface de usuário, dando-lhe opções para que possa, a partir da camada que definir, gerar tabelas diferentes de acordo com os critérios escolhidos e, posteriormente, estudar a que melhor representa as características da deformação frágil do maciço. Entre as vantagens obtidas com a utilização do *software* desenvolvido, destacam-se as seguintes:

### Possibilidade da escolha da área a ser analisada

O *AzimuthFinder* analisa todos os dados contidos em uma camada (*layer*) indicada pelo usuário. Entretanto, se este deseja restringir sua análise a uma área específica, ele pode gerar uma nova camada que contenha apenas parte dos lineamentos da camada fonte, utilizando-se, por exemplo, a ferramenta do tipo *Clip* (Figuras 3 a 5), inseridas nos *softwares* de geoprocessamento e que possibilitam o recorte de áreas irregulares para análises específicas. Desse modo é possível, por exemplo, gerar diversos diagramas específicos, que representem a malha estrutural contida em maciços individualizados ou em zona homóloga, como é o caso do Granito Serra do Tabuleiro (Figura 3), localizado próximo ao litoral catarinense e estudado por Reis (2011).

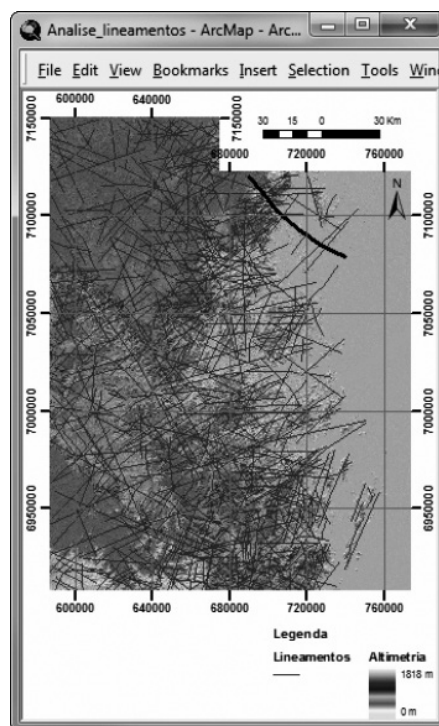


Figura 2. Mapa da área de estudo no ArcMap com o denso traçado de lineamentos em formato *shape* com o destaque para a Falha do Palmital.

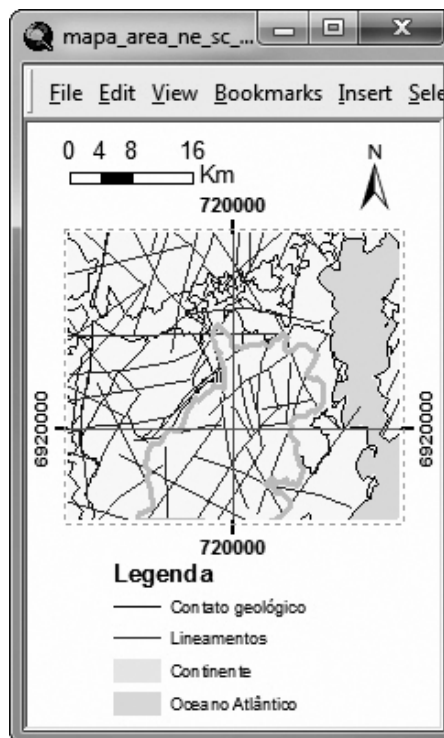


Figura 3. Individualização de lineamentos contidos em litotipo selecionado, no caso o Granito Serra do Tabuleiro.

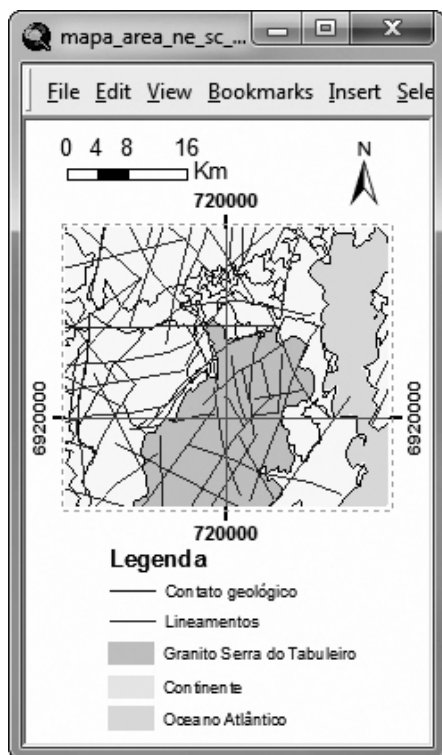


Figura 4. Visualização de nova camada a partir da seleção.

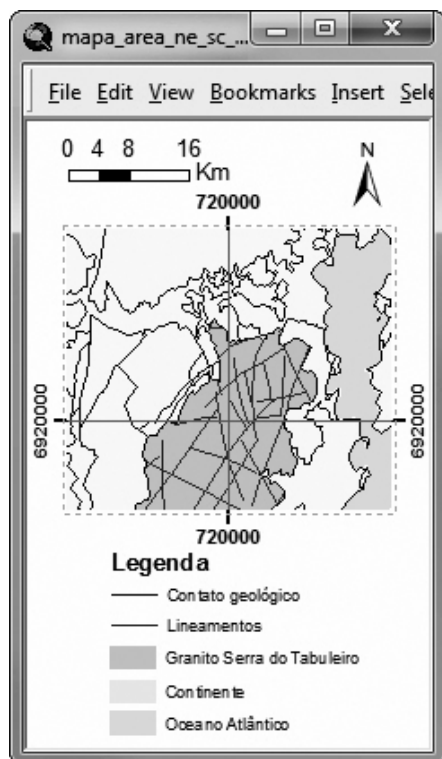


Figura 5. Recorte utilizando-se a ferramenta *Clip* para a geração de nova camada apenas com os lineamentos contidos no granito.

### Obtenção de dados numéricos azimutais

A rapidez e facilidade na obtenção da tabela que será gerada e exportada para os *softwares* específicos de análise estrutural é o maior ganho da ferramenta que foi desenvolvida. A celeridade do processo permite ao usuário estruturalista ter em mãos várias tabelas para a confecção de Diagramas de Rosetas, que manualmente seriam geradas por meio de demorada manipulação gráfica, com o risco da imprecisão. Assim, de forma rápida e controlada, diversos testes podem ser realizados com a confecção de diagramas, utilizando diferentes parâmetros e/ou áreas diferentes, o que permite uma análise estatística mais apurada.

### Escolha do comprimento do segmento e atribuição de pesos

Ao utilizar o método de medidas de “comprimento”, o programa calcula automaticamente o menor valor de comprimento de linha entre os traços da camada, e o sugere como sendo um valor de comprimento padrão (peso 1). Entretanto, se for do interesse do usuário, o valor do comprimento utilizado para analisar a representatividade das linhas (peso 1) pode ser atribuído arbitrariamente.

Por exemplo, na área analisada, a Falha do Palmital (Figura 6) representa uma estrutura mais alongada do que as suas possíveis subsidiárias conjugadas, portanto possui, por princípio, uma direção de ruptura mais representativa do que aquelas. O usuário terá a opção de melhor qualificar estruturas maiores como essa, dando-lhes a importância geológica mais condizente com o modelo de fraturamento, ou de deformação, que irá empregar posteriormente. Assim, há a possibilidade de construção de diferentes Diagramas de Rosetas, dando a opção de escolha daquele que reproduz de forma mais realista o padrão de lineamentos foto-interpretados, e seu significado geológico (Figuras 7 e 8).

Caso decida aplicar esse recurso, supondo que o usuário opte por utilizar o comprimento padrão sugerido pelo *AzimuthFinder*, ou seja, 3.101 m no estudo de caso para peso 1, as atitudes da Falha do Palmital (nordeste de Santa Catarina), que possui 64.975 m de comprimento no mapa, serão impressas no arquivo de texto 21 vezes ( $64.975/3.101 = 20,95$ ). Deve-se salientar que essa não é uma condição obrigatória do *software*, e sim está condicionada à escolha que o usuário desejar, pois se o comprimento escolhido fosse de 6.202 m, a atitude média seria impressa apenas 10 vezes.

### Liberdade na exportação de dados

Após escolher os outros parâmetros, o usuário deve indicar um dos *softwares* disponíveis (*Stereo32*, *Win Tensor*, *OpenStereo* e *Stereonet 7*) para que o *AzimuthFinder* crie

um arquivo de texto ou uma tabela (Figura 9) com a formatação que o programa indicado reconhece (Figura 10). Caso deseje exportar os dados para um *software* não indicado na lista, recomenda-se utilizar os arquivos de texto gerados para o programa *OpenStereo*, o qual possui entrada bastante simples, para adaptar o formato de acordo com as exigências do *software* escolhido.

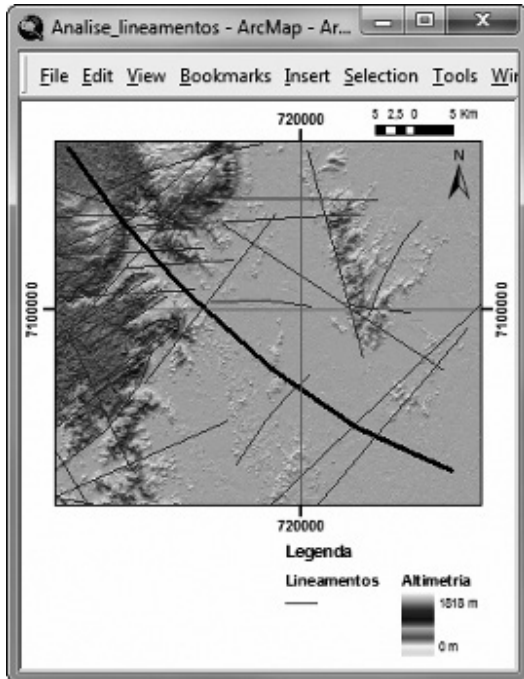


Figura 6. Falha do Palmital em destaque.

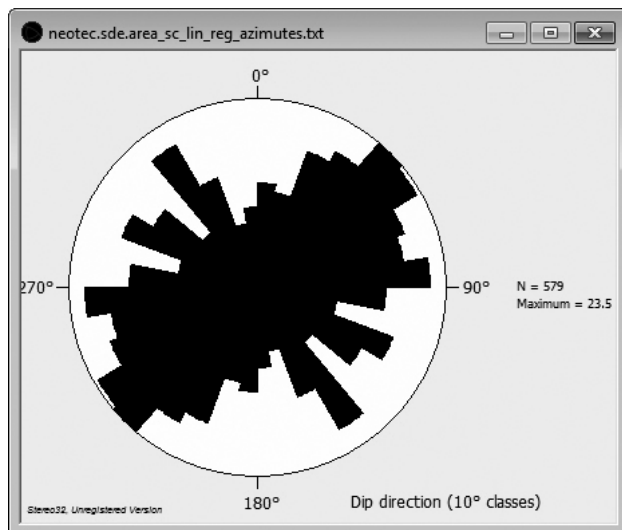


Figura 7. Estereograma do *software* Stereo32 gerado a partir de lineamentos traçados na área de estudo sem a atribuição de pesos (método de frequência), o que confere mesma representatividade entre lineamentos maiores e menores.

## Aplicação em outras análises

A obtenção de dados azimutais, a partir de feições lineares traçadas em ambiente SIG, pode ser utilizada também para outros tipos de análise, como para a confecção de Diagramas de Rosetas das direções gerais de segmentos de drenagens, que auxiliam na caracterização morfo-estrutural de área qualquer.

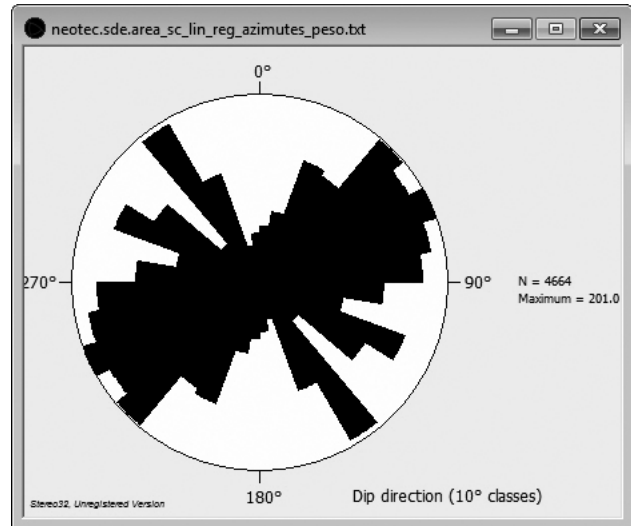


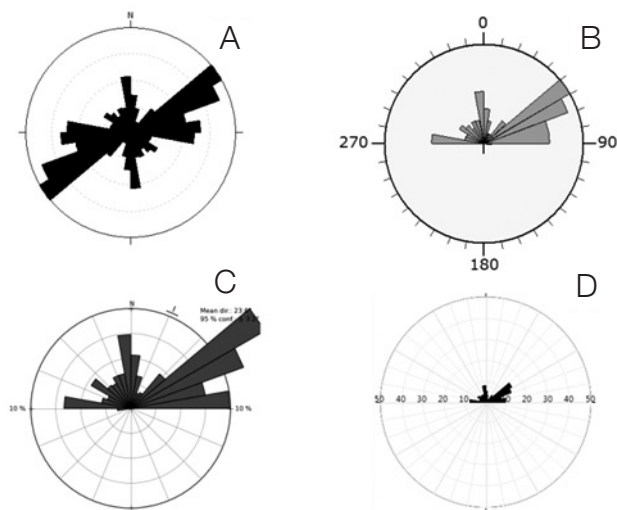
Figura 8. Estereograma do *software* Stereo32 gerado a partir de lineamentos traçados na área de estudo com a atribuição de pesos (método de comprimento), o que confere maior representatividade aos lineamentos mais longos (no exemplo utilizou-se o intervalo padrão sugerido pelo *AzimuthFinder*).

Stereo32 - Bloco de...		Stereonet 7 - Bloco ...	
Arquivo	Editar	Formatar	Exibir
Ajuda			
332	0	1	
69	0	1	
5	0	1	
75	0	1	
79	0	1	
70	0	1	
346	0	1	
68	0	1	
70	0	1	

OpenStereo - Bloco...		Win Tensor - Bloco ...	
Arquivo	Editar	Formatar	Exibir
Ajuda			
332	00	X	
69	00	X	
005	00	X	
075	00	X	
079	00	X	
070	00	X	
346	00	X	
068	00	X	
070	00	X	

Figura 9. Tabelas dos mesmos dados com quatro formatações diferentes no bloco de notas.



**Figura 10.** Diagramas de Rosetas dos mesmos dados gerados nos quatro softwares diferentes. Stereo32 (A), Win Tensor (B), OpenStereo (C) e Stereonet 7 (D). Observar que, a pesar da escala e formatos diferenciados, a diferença relativa entre direções é semelhante.

## CONCLUSÃO

O software denominado *AzimuthFinder* foi desenvolvido em linguagem *Python* para ser incorporado a título de aplicação ao conjunto de programas do ArcGis®. Para tanto, foi necessário o conhecimento de linguagem de programação, bem como o entendimento do comportamento estrutural frágil de maciços e as heterogeneidades reológicas que controlam sua densidade de fraturas.

Seu objetivo é resolver um problema prático da análise estrutural de lineamentos traçados em ambiente SIG, que é a obtenção automática de tabelas com direções azimutais necessárias à confecção de Diagramas de Rosetas para a interpretação de lineamentos estruturais, a partir de lineamentos estruturais foto-interpretados. Embora o Diagrama de Rosetas seja a representação de gráficos estatísticos mais simples, há dificuldade em obtê-lo a partir dos lineamentos traçados em meio digital, pois a obtenção de dados azimutais geralmente é trabalhosa, posto que demanda várias etapas de processamento de dados.

O *AzimuthFinder* soluciona essa dificuldade porque mede espacialmente os alinhamentos (ou lineamentos) traçados em imagens em relação ao norte verdadeiro, e constrói de forma automática as tabelas necessárias à confecção de Diagramas de Rosetas. O programa também permite que sejam geradas várias tabelas para o mesmo objeto de análise, utilizando diferentes pesos específicos e/ou áreas.

O programa exporta os dados para um arquivo de formato TXT, que é o formato mais simples de entrada de dados para vários programas de análise estrutural, mas foi

preparado para gerar saída de exportação específica com o formato de entrada de dados dos softwares *Stereo32*, *Win Tensor*, *OpenStereo* e *Stereonet 7*. Os testes com o programa demonstraram que o *AzimuthFinder* é bastante eficiente e rápido, facilitando a análise estrutural em áreas submetidas à deformação frágil.

## AGRADECIMENTOS

Os autores agradecem ao Prof. Dr. Elvo Fasbinder pelas discussões críticas iniciais; à Dandara Salvador, bolsista de Iniciação Científica do programa de Graduação em Geologia da UFPR pelo auxílio nos testes do programa, e finalmente aos revisores indicados pela revista.

## REFERÊNCIAS

- Allmendinger, R. W. (2011). *Stereonet*© 7.3.0. Cambridge, England.
- Bettú, D. F. (2012). *Novo método probabilístico para mapeamento de unidade de paisagem geomórficas - caracterização geoambiental das bacias hidrográficas dos rios Passaúna e Barigui*. Tese (Doutorado). Paraná: Departamento de Geologia – Universidade Federal do Paraná (UFPR).
- Delvaux, D. (2013). *Win Tensor 4.0.4*. Royal Museum for Central Africa, Dept. Geology & Mineralogy. Tervuren, Belgium.
- Economic and Social Research Institute (ESRI). ArcGIS® Resources. Acessado em 25 de junho de 2012, <<http://resources.arcgis.com>>.
- Economic and Social Research Institute (ESRI). Archived Forums. Acessado em 25 de junho de 2012, <<http://forums.esri.com>>.
- Economic and Social Research Institute (ESRI). Esri Developer Network. Acessado em 25 de junho de 2012, <<http://edn.esri.com/>>.
- Fonsêca, D. N., Corrêa, A. C. B. (2011). Uso de MDE na extração de lineamentos para detecção de reativações neotectônicas na Bacia do Rio Preto, Serra do Espinhaço Meridional, MG. *Simpósio Brasileiro de Sensoriamento Remoto*, 15. 3483 – 3490. Curitiba: Anais.
- Gama, C. D. (1982). Computação gráfica interativa de dados geológicos vetoriais. *Simpósio Brasileiro de Sensoriamento Remoto*, 2. v.1, 17-24. Brasília: Anais.



- Gillespie, P. A., Howard, C. B., Walsh, J. J., Watterson, J. (1993). Measurement and characterization of spatial distributions of fractures. *Tectonophysics*, 226, 113-141.
- Grohmann, C. H., Campanha, G. A. C. (2010). Open Stereo: open source, cross-platform software for structural geology analysis. *AGU 2010 Fall Meeting*. San Francisco, CA.
- Guo, C., George, S. A., Lindsey R. P. (1999). Statistical analysis of surface lineaments and fractures for characterizing naturally fractured reservoirs. In: R. Schatzinger, J. Jordan (Eds.), *Reservoir Characterization-Recent Advances AAPG Memoir* (v.71, 221-250).
- Haman, P. J. (1961). Lineament analysis on aerial photographs exemplified, in the North Sturgeon Lake area, Alberta. *Calgary: West-Canadian Research publ*, 2 (1), 23.
- Huntington, J. F. (1969). Methods and applications of fracture trace analysis in the quantification of structural geology. *Geological Magazine*, 106(5), 430-451.
- Justo, P. P. (2006). *Sistemas orbitais e aéreos aplicados à análise multi-escala de lineamentos na borda sudeste da Bacia do Parnaíba*. Dissertação (Mestrado). Natal: Centro de Ciências Exatas e da Terra – UFRN.
- Liu, C. C. (1987). A geologia estrutural do Estado do Rio de Janeiro vista através de imagens MSS de Landsat. *Simpósio de Geologia Regional RJ - ES*, v.1, 164-188. Rio de Janeiro: SBG.
- Liu, C. C. (1984). Análise estrutural de lineamentos em imagens de Sensoriamento Remoto: aplicação do Estado do Rio de Janeiro. Tese (Doutorado em Geologia). São Paulo: Universidade de São Paulo –USP.
- O’Leary, D. W., Friedman, J. D., Pohn, H. A. (1976). Lineaments, linear, lineations: some proposed new standards for old terms. *Geological Society of America Bulletin*, 87, 1463-1469.
- Paradella, W. R., Vitorello, I. (1981). Application of computerized techniques using Landsat images for geological studies. *COGEODATA IAMG meeting for South America*, v.1, 1-26. Rio de Janeiro: Anais.
- Reis, J. P. P. (2011). *Morfotectônica da Serra do Tabuleiro*. Dissertação (Mestrado em Programa de Pós-Graduação em Geografia). Santa Catarina: Universidade Federal de Santa Catarina – UFSC.
- Riccomini, C., Crósta, A. P. (1988). Análise preliminar de lineamentos em imagens de sensores remotos aplicadas à prospecção mineral na área dos granitóides Mandira, SP. *Boletim IG – USP, Série Científica*, 19, 23-37.
- Roller, K., Trepmann, C. A. (2003). Stereo32 1.0.3. Institut für Geologie - Ruhr University Bochum: Bochum, DE.
- Santos A. R. Dos. (1986). O sensoriamento remoto e a estrutura rúptil: exemplo de aplicações em estudos neotectônicos. *Simpósio Latino-Americano de Sensoriamento Remoto*, v. 1, 429-434. Gramado (RS).
- Soares, P. C., Fiori, A. P. (1976). Lógica e sistemática na análise e interpretação de fotografias aéreas em geologia. *Notícia Geomorfológica*, 16(32), 71–104.
- Strieder, A. J., Amaro, V. E. (1997). Estruturas de lineamentos extraídos de imagens de sensores remotos. EGATEA. *Revista da Escola de Engenharia*, 25(4), 109-117.
- Taylor, G. R. (1988). Image analysis techniques for the interpretation of airphoto lineaments – petroleum exploration, Eromanga Basin, Australia. *Geocarto International*, 3, 53-60.
- Yamaguchi, Y. (1985). Image-scale and look-direction effects on the detectability of lineaments in radar images. *Remote Sensing of Environment*, 17, 117-127.

**Anexo 1.** Instalação do AzimuthFinder no ArcMap: para instalar o AzimuthFinder o usuário deve inicialmente abrir o ArcGIS. Em seguida abrir a *toolbox* e então escolher o comando para importar ferramenta (*Add Toolbox...*), após esse procedimento selecionar o arquivo de extensão PY (*AzimuthFinder.py*). Se desejar manter sempre o AzimuthFinder na *toolbox*, deve-se escolher o comando de salvar as preferências (*Save Settings > To Default*).

**Anexo 2.** Código AzimuthFinder.py.

```
# -*- coding: cp1252 -*-
# -----
# AzimuthFinder.py
# Criado em: 19/03/2013
# Autor: Gustavo Lopes Queiroz
# Contato: gustavo.lopes.queiroz@gmail.com
# Agradecimentos a: Ednilson Nascimento, coautor do programa; e Eduardo
# Salamuni, professor orientador do programa de iniciação científica
# -----

# Importarmodulos
import sys, string, os, arcgisscripting, math, datetime

# Geoprocessor object
gp = arcgisscripting.create()

# Entrada de dados
Lineamentos = gp.GetParameterAsText( 0 )
UsarPeso = gp.GetParameterAsText( 1 )
TamanhoUnidPeso = gp.GetParameterAsText( 2 )
MergulhoArbitr = gp.GetParameterAsText( 3 )
SalvarPara = gp.GetParameterAsText( 4 )
PastaDestino = gp.GetParameterAsText( 5 )
NomeSaida = gp.GetParameterAsText( 6 )

# Permitir que o processamento sobrescreva arquivos antigos
gp.overwriteOutput = True

# Identificar o campo de geometria
DescLayer = gp.Describe( Lineamentos )
CampoGeometria = DescLayer.ShapeFieldName

defComprimento( LinhaPointer ):
    Feature = LinhaPointer.GetValue( CampoGeometria )
    return Feature.Length

defObterListPont( Linha ):
    Feature = Linha.GetValue( CampoGeometria )
    ListaPontos = []
    ParteNum = 0
    ParteTotal = Feature.PartCount
    # Loop para cada parte do feature
    whileParteNum<ParteTotal:
        Parte = Feature.GetPart( ParteNum )
        Vertice = Parte.Next()
        whileVertice:
            ifVertice:
```

Continua...

## Anexo 2. Continuação.

```

# Adicionar as coordenadas de cada vértice à lista de pontos
ListaPontos.append( [ Vertice.X, Vertice.Y ] )
Vertice = Parte.Next()

ParteNum += 1
returnListaPontos

def sign( integer ):
if integer != 0:
    returnint( integer / abs( integer ) )
else:
return 0

try:

    #Criararquivo TXT
extension = ".txt"
txtfile = open( PastaDestino + "\\\" + NomeSaida.replace( "." , "" ) + extension, 'w' )

#Linhas de cabeçalho
ifSalvarPara == "Stereonet 7":
txtfile.write( "TP\n" )
ifSalvarPara == "Win Tensor":
txtfile.write( "Text (Tab delimited) Win_Tensor Data File :
    " + PastaDestino + "\\\" + NomeSaida.replace( "." , "" ) + extension + "\n\n
Project code                                \n Outcrop code
                                \n Area name                                \n Geologist
                                \n Field date
                                \n Incoding date                                " +
str(datetime.date.today().day) + "/" + str(datetime.date.today().month) + "/" +
str(datetime.date.today().year) + " \n Magnetic North
0                                \n Latitude                                \n Longitude
                                \n Outcrop Quality
                                \n Outcrop Stratigraphy                                \n Outcrop
Lithology                                \n Outcrop Structure
                                \n Supposed age of movement
                                \n Half dieder angle p axis                                \n Input mode
                                11 \n\n Fault-slip data

Input row data (User format)
    Compiled data Set 1 (Standard format)
    Compiled data Set 1 (Standard format)
    Kinematic axes                                WSM Regime
                                \n Properties                                Orientation 1 Orientation 2
                                Slip Conf. Weight Activ. Striae Subset Comments Fault Plane
Slip Line                                Slip Subsets Fault Plane Slip Line
Slip Subsets                                P P B B T T SHmax Shmin
Regime Regime\n Id Format Type Ori1A Ori1B Ori2A Ori2A Sense Level
Factor Type Intens. Index Dip Dip-Dir Azim. Plunge Sense
Temp. Final Dip Dip-Dir Azim. Plunge Sense Temp. Final Azim.
Incl. Azim. Incl. Azim. Incl. Azim. Azim. Code Index\n" )

ID = 1

CursorLn = gp.SearchCursor( Lineamentos )
LinhaLn = CursorLn.Next()

whileLinhaLn:
ListVert = ObterListPont( LinhaLn )

    #Trigonometria

X = ListVert[ 0 ][ 0 ]
Y = ListVert[ 0 ][ 1 ]
XX = ListVert[ -1 ][ 0 ]
YY = ListVert[ -1 ][ 1 ]

CatetoA = abs( Y - YY )

```

Continua...

## Anexo 2. Continuação.

```

CatetoB = abs( X - XX )

ifCatetoB == 0:
CatetoB = 0.00001

if ( sign( X - XX ) * sign( Y - YY ) ) == 1:
Azimute = 90 - int(math.degrees( math.atan( CatetoA / CatetoB ) ) )
else:
Azimute = 270 + int(math.degrees( math.atan( CatetoA / CatetoB ) ) )

#Escrever no arquivo de texto os azimutes linha a linha

ifUsarPeso == "Frequência":
    Repetir = 1
else:
    Repetir = round( Comprimento( LinhaLn ) / long( TamanhoUnidPeso ) )

ifRepetir < 1:
Repetir = 1

whileRepetir > 0:
ifSalvarPara == "Stereo32":
txtfile.write(str( Azimute ) + "\t" + MergulhoArbitr + "\t" + "1" + "\n" )
elifSalvarPara == "Stereonet 7":
txtfile.write(str( Azimute ) + "\t" + MergulhoArbitr + "\t" + "\n" )
elifSalvarPara == "OpenStereo":
txtfile.write(str( Azimute ) + "\n" )
elifSalvarPara == "Win Tensor":
print_azimuth = str(Azimute)
while (len(print_azimuth)<3):
print_azimuth = "0" + print_azimuth
print_dip = MergulhoArbitr
while (len(print_dip)<2):
print_dip = "0" + print_dip
txtfile.write(str(ID) + "      22      8          " + print_azimuth + " " +
print_dip + " X      X      1,0      0      0      1,0          " +
print_dip + " " + print_azimuth + " XA      1,0

      UA      14\n" )
ID += 1
      Repetir -= 1

LinhaLn = CursorLn.Next()

#Linhas de rodapé
ifSalvarPara == "Win Tensor":
txtfile.write( "\nBegin Subset Properties\nIndex      Label      Color Code
      Selected\nSet_01      0      -1\nSubset_01,0      0      -1\nEnd
Subset Properties\nBegin Solution Properties\nSolution name      Method Sigma1 Sigma2
      Sigma3 R      Psi      Subsets Comments      Init_Data      Process_Param\nCurrent
solution      1,0      \nEnd
Solution Properties\n" )

# SAÍDA
txtfile.close()
gp.AddMessage( "Processamento concluído com sucesso!" )

except:
gp.AddError( "Erro!" )

```