

APÊNDICE I – PROGRAMAS EM LEGAL

Programa I.1 – Programa de reclassificação das unidades litológicas em classes favorável e não-favorável.

```
{//declaração
Tematico      lito, favo ("Litologia");
Tabela        tabfav (Reclassificacao);

//tabela
tabfav = Novo (CategoriaIni = "Litologia" ,
              CategoriaFim = "Litologia",
              "embasamento" :      "nao-favoravel",
              "arenitos" :          "nao-favoravel",
              "fonolitos" :         "nao-favoravel",
              "tinguaitos" :        "nao-favoravel",
              "foiaitos" :          "favoravel",
              "R_potassicas" :      "favoravel",
              "lujaritos/chibin" :  "favoravel",
              "mat.vulcanico" :     "favoravel" );

//instanciação
lito = Recuperar (Nome = "Litologia");
favo = Novo (Nome = "Litologia favorável: contrast", ResX = 25, ResY = 25, Escala = 50000);

//operação
favo = Reclassifique (lito, tabfav);}
```

Programa I.2 - Programa de reclassificação das unidades de intensidade radioativa em classes favorável e não-favorável.

```
{//declaração
Tematico      favo , gama ("Gama");
Tabela        tabfav (Reclassificacao);

//tabela
tabfav = Novo (CategoriaIni = "Gama" ,
              CategoriaFim = "Gama",
              "background < 1.3" :  "nao-favoravel",
              "1.3 <x< 1.8" :       "nao-favoravel",
              "1.8 <x< 2.5" :       "favoravel",
              "2.5 <x< 3.5" :       "favoravel",
              "x>3.5" :             "favoravel");

//instanciação
gama = Recuperar (Nome = "Gama");
favo = Novo (Nome = "Gama favorável", ResX = 25, ResY = 25, Escala = 50000);
//operação
favo = Reclassifique (gama, tabfav);
}
```

Programa I.3 – Programa de integração das evidências através de operadores lógicos booleanos

```
{//declaração
Tematico      lito ("Litologia"), cont ("Contato-geológico"), circ ("Estruturas-Circulares"),
line ("Lineamentos"), gama ("Gama"), boo ("Integração - Resultados");

//Instanciação
lito = Recuperere (Nome = "Litologia-favorável");
cont = Recuperere (Nome = "Buffer: 100m");
circ = Recuperere (Nome = "Buffer: 350m");
line = Recuperere (Nome = "Buffer: 250m");
gama = Recuperere (Nome = "Gama-favorável");
boo = Novo (Nome = "Booleano", ResX = 25, ResY = 25, Escala = 50000);

//Operação
boo = (((lito.Classe == "favoravel" || cont.Classe == "favoravel") && (circ.Classe == "favoravel" ||
line.Classe == "favoravel") && gama.Classe == "favoravel") ? Classe ("favoravel") : Classe("nao-
favoravel"));
}
```

Programa I.4 – Programa para ponderação das classes litológicas

```
{//declaração
Tematico      lito ("Litologia");
Numerico      pond ("Média Ponderada");
Tabela        tablito (Ponderacao);

//tabela
tablito = Novo (CategoriaIni = "Litologia",
"arenitos", "embasamento", "fonolitos", "tinguaitos" : 0,
"mat.vulcanico" : 20,
"foiaitos" : 30,
"R_potassicas", "lujaritos/chibin" : 60);

//instanciação
lito = Recuperere (Nome = "Litologia");
pond = Novo (Nome = "Litologia ponderada", ResX = 25 , ResY = 25 , Escala =50000, Min = 0, Max = 100);

//operação
pond = Pondere (lito , tablito);
}
```

Programa I.5 – Programa para ponderação das classes de intensidade gama-radiométrica

```
{//declaração
Tematico      gama ("Gama");
Numerico      numg ("Média Ponderada ");
Tabela        tabgama (Ponderacao);
//tabela
tabgama = Novo (CategoriaIni = "Gama",
"background" : 0.0,
"1.3-1.8" : 10,
"1.8-2.5" : 60,
```

```

"2.5-3.5" : 70,
">3.5" : 80);
gama = Recuperar (Nome = "Gama");
numg = Novo (Nome = "Gama ponderado", ResX = 25 , ResY = 25 , Escala = 50000, Min = 0, Max = 100);

//operação
numg = Pondere (gama , tabgama);
}

```

Programa I.6 – Programa para união e ponderação dos “buffers” dos lineamentos e estruturas circulares.

```

{//declaração
Tematico      line, line-circ ("Lineamentos"), circ ("Estruturas-Circulares");
Numerico      line-circ-pond, ("Média Ponderada");
Tabela        tabline-circ (Ponderacao);

//instanciação
line = Recuperar (Nome = "Buffer: 250m");
circ = Recuperar (Nome = "Buffer: 350m");
line-circ = Novo (Nome = "Lineam-estr.circular", ResX = 25,ResY = 25 ,Escala =50000);
line-circ-pond = Novo (Nome = "Lineam-estr.circular ponderada", ResX = 25,ResY = 25 ,Escala =50000,
Min = 0, Max =100);

//tabelas
tabline-circ = Novo (CategorialIni = "Lineamentos ",
"nao-favoravel" : 0,
"favoravel" : 20);

//operação
line-circ = ((line.Classe == "favoravel" || circ.Classe == "favoravel") ? Classe ("favoravel") : Classe ("nao-favoravel"));

line-circ-pond = Pondere (line-circ , tabline);
}

```

Programa I.7 – Programa de integração dos planos de informação ponderados através de uma soma ponderada.

```

{//declaração
Numerico      lito, gama, line-circ, result ("Média Ponderada");

//instanciação
lito = Recuperar (Nome = "Litologia ponderada");
gama = Recuperar (Nome = "Gama ponderada");
line-circ = Recuperar (Nome = "Lineam-estr.circular ponderada");
result = Novo (Nome = "Integração-M.ponderada", ResX = 25 , ResY = 25 , Escala = 50000, Min = 0, Max = 100);
//operação
result = ((gama*80) + (lito*60) + (line-circ*20))/160;
}

```

Programa I.8 – Programa de ponderação das classes de anomalia gama-radiometria para obtenção dos membros *fuzzy*.

```
{//declaração
Tematico      gama ("Gama");
Numerico      numg ("numfuzzy");
Tabela tabgama (Ponderacao);

//tabela
tabgama = Novo (CategoriaIni = "Gama",
"background" : 0.0,
"1.3-1.8" : 0.125,
"1.8-2.5" : 0.750,
"2.5-3.5" : 0.875,
"> 3.5" : 1.000);

//instanciação
gama = Recuperar (Nome = "Gama");
numg = Novo (Nome = "Gama-fuzzy", ResX = 25 , ResY = 25 , Escala = 50000, Min = 0, Max = 1);

//operação
numg = Pondere (gama , tabgama);
}
```

Programa I.9 – Programa para construção dos membros fuzzy da rocha potássica

```
{// programa para espacialização continua e rígida do contato da R. potássica
// declaração
Tematico      masC ("Contato-geológico"), lito ("Litologia") ;
Numerico      dist ("Mapa de distância") , result ("numfuzzy");

// instanciação
masC = Recuperar (Nome = "Máscara:corpos pequenos");
lito = Recuperar (Nome = "Litologia-R.potássica");
dist = Recuperar (Nome = "contato-R.potássica");
result = Novo (Nome = "R-potássica: fuzzy", ResX = 25, ResY = 25, Escala = 50000 , Min = 0 , Max = 1);

// operação
result = ((masC.Classe == "favoravel" && lito.Classe == "R_potassicas") ? Digital (1) :
(masC.Classe == "favoravel" && lito.Classe == "nao-favoravel") ? Digital (0) :
(lito.Classe == "R_potassicas" && dist >= 100) ? Digital (1) :
(dist <100 && lito.Classe == "R_potassicas") ? (dist*0.005)+0.5 :
(dist < 100 && lito.Classe == "nao-favoravel") ? (dist* -0.005)+0.5 : Digital (0));
}
```

Programa I.10 – Programa para construção dos membros fuzzy do Lujarito/chibinito

```
{// programa para espacialização continua e rígido do contato do lujarito/chibinito
// declaração
Tematico      masC ("Contato-geológico"), lito ("Litologia") ;
Numerico      dist ("Mapa de distância") , result ("numfuzzy");

// instanciação
masC = Recuperar (Nome = "Máscara:corpos pequenos");
```

```

lito = Recupere (Nome = "Litologia - lujarito/chibinito");
dist = Recupere (Nome = "Cont.geológico - lujarito/chibin");
result = Novo (Nome = "Lujarito/chibinito: fuzzy1", ResX = 25 , ResY = 25 , Escala = 50000 , Min = 0 ,
Max = 1);

// operação
result = ((masC.Classe == "favoravel" && lito.Classe == "lujaritos/chibin") ? Digital (1) :
(masC.Classe == "favoravel" && lito.Classe == "nao-favoravel") ? Digital (0) :
(lito.Classe == "lujaritos/chibin" && dist >= 100) ? Digital (1) :
(dist < 100 && lito.Classe == "lujaritos/chibin") ? (dist*0.005)+0.5 :
(dist < 100 && lito.Classe == "nao-favoravel") ? (dist* -0.005)+0.5 : Digital (0));
}

```

Programa I.11 – Programa para construção dos membros fuzzy do foiaíto

```

{// programa para espacialização continua e rígida dos contatos do foiaíto
// declaração
Tematico      masC ("Contato-geológico"), lito ("Litologia");
Numerico      dist ("Mapa de distância"), result ("numfuzzy");

// instanciação
masC = Recupere (Nome = "Máscara:corpos pequenos");
lito = Recupere (Nome = "Litologia-Foiaíto");
dist = Recupere (Nome = "Contato - Foiaíto");
result = Novo (Nome = "Foiaíto-fuzzy", ResX = 25 , ResY = 25 , Escala = 50000 , Min = 0 , Max = 1);

// operação
result = ((masC.Classe == "favoravel" && lito.Classe == "foiaitos") ? Digital (1) :
(masC.Classe == "favoravel" && lito.Classe == "nao-favoravel") ? Digital (0) :
(lito.Classe == "foiaitos" && dist >= 100) ? Digital (1) :
(dist < 100 && lito.Classe == "foiaitos") ? (dist*0.005)+0.5 :
(dist < 100 && lito.Classe == "nao-favoravel") ? (dist* -0.005)+0.5 : Digital (0));
}

```

Programa I.12 – Programa para construção dos membros fuzzy do material vulcânico

```

{// programa para espacialização continua e rígido do contato do tinguaito
// declaração
Tematico      masC ("Contato-geológico"), lito ("Litologia");
Numerico      dist ("Mapa de distância"), result ("numfuzzy");

// instanciação
masC = Recupere (Nome = "Máscara:corpos pequenos");
lito = Recupere (Nome = "Litologia- Mat. Vulcânico");
dist = Recupere (Nome = "Contato - Mat. vulcânico");
result = Novo (Nome = "Mat. Vulcânico: fuzzy", ResX = 25 , ResY = 25 , Escala = 50000 , Min = 0 , Max
= 1);

// operação
result = ((masC.Classe == "favoravel" && lito.Classe == "mat.vulcanico") ? Digital (1) :
(masC.Classe == "favoravel" && lito.Classe == "nao-favoravel") ? Digital (0) :
(lito.Classe == "mat.vulcanico" && dist >= 100) ? Digital (1) :
(dist < 100 && lito.Classe == "mat.vulcanico") ? (dist*0.005)+0.5 :
(dist < 100 && lito.Classe == "nao-favoravel") ? (dist* -0.005)+0.5 : Digital (0));
}

```

```
}
```

Programa I.13 – Programa para integração dos PI das unidades litológicas *fuzzy* através de uma soma ponderada.

```
{// programa para integração das litologias fuzzy
// declaração
Numerico      foia , luja , matvul, pota, result ("numfuzzy");

// instanciação
foia = Recuperar (Nome = "Foiaíto: fuzzy");
matvul = Recuperar (Nome = "Mat. Vulcânico: fuzzy");
pota = Recuperar (Nome = "R-potássica: fuzzy");
luja = Recuperar (Nome = "Lujarito/chibinito: fuzzy");
result = Novo (Nome = "Litologia-fuzzy", ResX = 25 , ResY = 25 , Escala = 50000 , Min = 0 , Max =2);

// Operação
result = (matvul*0.3333) + (foia*0.5) + pota + luja ;
```

Programa I.14 – Programa de mapeamento dos membros *fuzzy* das estruturas circulares.

```
{//Considerando um ponto cruzamento de350m
//declaração
Numerico est ("Mapa de distância") , fuzzy ("Evidências Fuzzy");

//instanciação
est = Recuperar (Nome = "Estrutura circular");
fuzzy = Novo (Nome = "Estr.circular-fuzzy-pc-350m" , ResX = 25, ResY = 25, Escala = 50000,
             Min = 0, Max = 1);

//operação
fuzzy = ( est <= 700 ? 1 / ( 1 +((1/122500)*(est^2) ) ) : Digital (0));
}
```

Programa I.15 – Programa de mapeamento dos membros *fuzzy* dos lineamentos

```
{//Considerando um buffer de interesse de 250m
//declaração
Numerico      line ("Mapa de distância") , fuzzy ("numfuzzy");

//instanciação
line = Recuperar (Nome = "Lineamentos");
fuzzy = Novo (Nome = "Lineamentos-fuzzy" , ResX = 25, ResY = 25, Escala = 50000,
             Min = 0, Max = 1);

//operação
fuzzy = ( line <= 250 ? 1 / ( 1 +(0.000064*(line^2) ) ) : Digital (0));
}
```

Programa I.16 – Programa de mapeamento dos membros *fuzzy* dos contatos dos corpos do foiaíto.

```
{//Considerando um buffer de interesse de 100m
```

```

//declaração
Numerico      cont ("Mapa de distância") , fuzzy ("numfuzzy");

//instanciação
cont = Recupere (Nome = "Contato - Foiaíto");
fuzzy = Novo (Nome = "Contato-foiaíto-fuzzy" , ResX = 25, ResY = 25, Escala = 50000,
             Min = 0, Max = 1);

//operação
fuzzy = ( cont <= 100 ? 1 / ( 1 +(0.0004*(cont^2) ) ) : Digital (0));
}

```

Programa I.17 – Programa de integração dos PI *fuzzy* (evidências) segundo operadores *fuzzy* mínimo-máximo.

```

{ //declaração
Numerico  gama, lito, circ, cont, line, maxi1, maxi2 ("numfuzzy");
Numerico  minfu ("Fuzzy-resultados");

//instanciação
gama = Recupere (Nome = "Gama-fuzzy");
line = Recupere (Nome = "Lineamentos-fuzzy");
cont = Recupere (Nome = "Contato-foiaíto-fuzzy");
lito = Recupere (Nome = "Litologia-fuzzy");
circ = Recupere (Nome = " Estr-circular-500-300-fuzzy");
minfu = Novo (Nome = "Fuzzy:min-max", ResX = 250, ResY = 250, Escala = 50000,
             Min= 0, Max = 1 );
= Novo (Nome = "f11", ResX = 25, ResY = 25, Escala = 50000,
       Min= 0, Max = 1 );
maxi2 = Novo (Nome = "f12", ResX = 25, ResY = 25, Escala = 50000,
             Min= 0, Max = 1 );

//operação
maxi1 = max (lito, cont);
maxi2 = max (circ, line);
minfu = min (gama, min (maxi1 , maxi2));
}

```

Programa I.18 – Programa de integração dos PI *fuzzy* (evidências) segundo operador *fuzzy* média.

```

{ //declaração
Numerico      gama, lito, circ, cont, line ("numfuzzy");
Numerico  minfu ("Fuzzy-resultados");

//instanciação
gama = Recupere (Nome = "Gama-fuzzy");
line = Recupere (Nome = "Lineamentos-fuzzy");
cont = Recupere (Nome = "Contato-foiaíto-fuzzy");
lito = Recupere (Nome = "Litologia-fuzzy");
circ = Recupere (Nome = " Estr-circular-500-300-fuzzy");
minfu = Novo (Nome = "Fuzzy-media", ResX = 25, ResY = 25, Escala = 50000,
             Min= 0, Max = 1 );
}

```

```
//operação
minfu = (gama +lito +circ + cont + line)/5;
}
```

Programa I.19 – Programa de integração dos PI *fuzzy* (evidências) segundo operador *fuzzy* ponderado.

```
{//declaração
Numerico      gama, lito, circ, cont, line ("numfuzzy");
Numerico      minfu ("Fuzzy-resultados");

//instanciação
gama = Recupere (Nome = "Gama-fuzzy");
line = Recupere (Nome = "Lineamentos-fuzzy");
cont = Recupere (Nome = "Contato-foiaíto-fuzzy");
lito = Recupere (Nome = "Litologia-fuzzy");
circ = Recupere (Nome = " Estr-circular-500-300-fuzzy");
minfu = Novo (Nome = "Fuzzy-ponderado", ResX = 25, ResY = 25, Escala = 50000,
Min= 0, Max = 1 );

//operação
minfu = (gama*0.514) +(lito*0.258) +(circ*0.1223) + (cont*0.0529) + (line*0.0529);
}
```

Programa I.20 – Programa de integração dos PI *fuzzy* (evidências) segundo operador *fuzzy* gama.

```
{//declaração
Numerico      gama, lito, circ, cont, line, sum, pro ("numfuzzy");
Numerico      result ("Fuzzy-resultados");

//instanciação
gama = Recupere (Nome = "Gama-fuzzy");
line = Recupere (Nome = "Lineamentos-fuzzy");
cont = Recupere (Nome = "Contato-foiaíto-fuzzy");
lito = Recupere (Nome = "Litologia-fuzzy");
circ = Recupere (Nome = " Estr-circular-500-300-fuzzy");
sum = Novo (Nome = "sumfuzzy", ResX = 25, ResY = 25, Escala = 50000,
Min = 0, Max = 1);
pro = Novo (Nome = "profuzzy", ResX = 25, ResY = 25, Escala = 50000,
Min = 0, Max =1 );
result = Novo (Nome = "Fuzzy-Gama (0.85)", ResX = 25, ResY = 25, Escala = 50000,
Min = 0, Max = 1);

//operação
//soma algébrica fuzzy
sum = 1-((1-lito)*(1-gama)*(1-circ)*(1-line)*(1-cont));

//produto algébrico fuzzy
pro = lito*gama*circ*line*cont;

//valor do gama
g = 0.85;
```

```
//operador gama
result = (sum^g)*(pro^(1-g));
}
```

Programa I.21 – Programa para cálculo da chance a posteriore

```
{//declaração
Tematico      gama ("Gama"), line ("Lineamentos"), cont ("Contato-geológico"),
              circ ("Estruturas-Circulares"), lito ("Litologia");
Numerico      logO , prob ("Bayes");

//atribuição dos valores dos W+ e W-
Wpri = -4.36964279;
WposGama = 1.685188964; WnegGama = -0.704475879;
WposLine = 0.149913968; WnegLine = -0.103384395;
WposCont = 0.798634581; WnegCont = -0.187620352;
WposCirc = 0.898897373; WnegCirc = -0.401959283;
WposLito = 1.253996113; WnegLito = -0.454511882;

//Instanciação
gama = Recupere (Nome = "Gama-favorável");
line = Recupere (Nome = "Buffer-250m");
cont = Recupere (Nome = "Buffer: 500m");
circ = Recupere (Nome = "Buffer: int-500m-ext-300m");
lito = Recupere (Nome = "Lito-favorável");
logO = Novo (Nome = "LN(o(d|p))", ResX = 25, ResY =25, Escala = 50000, Min= -10, Max = 1);
prob = Novo (Nome = "prob-datadriven", ResX = 25, ResY =25, Escala = 50000, Min= 0, Max = 1);

//operação
logO = Wpri +
(( line.Classe == "favoravel" ) ? Digital ( WposLine) : Digital(WnegLine))+
(( gama.Classe == "favoravel" ) ? Digital ( WposGama) : Digital ( WnegGama)) +
(( cont.Classe == "favoravel" ) ? Digital ( WposCont) : Digital (WnegCont))+
(( circ.Classe == "favoravel" ) ? Digital ( WposCirc) : Digital (WnegCirc))+
(( lito.Classe == "favoravel" ) ? Digital ( WposLito) : Digital (WnegLito));

prob = exp (logO) / ( 1 + exp (logO));
}
```