# **IDL Tutorial**

AULA 11

**PROJETO** 

- Montar um **projeto** de um simulador do modelo Okumura Hata dividido em pequenos blocos.
- O projeto deverá ser organizado em um **Script**.
- Deverá ser dividido em algumas sub-rotinas, e algumas funções obedecendo as divisões solicitadas a seguir.

## • Subrotina 1:

 Simula o modelo Okumura Hata e chama subrotina que salva os resultados.

### • Subrotina 2:

o Salva os valores em um arquivo 'dados.txt'

## Subrotina 3:

o Ler arquivo 'dados.txt' e compartilha as variáveis dos dados.

## Subrotina 4:

 Ler arquivos compartilhados e calcula média, número de positivos e número de negativos através de funções, e chama subrotina para gerar mini-relatório.

## Subrotina 5:

 Salva arquivo com mini-relatório, com os dados da simulação, a média, o número de negativos e o número de positivos.

### • Subrotina 6:

o Gera gráfico dos dados simulados.

### • Função 1:

O Calcula a perda de potencial seguindo o modelo Okumura Hata.

## • Função 2:

o Calcula a média de uma série de dados.

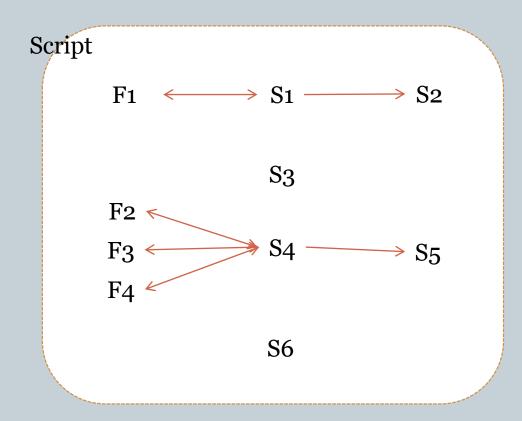
## Função 3:

o Calcula o número de positivos de uma série de números.

### • Função 4:

O Calcula o número de negativos de uma série de números.

## Fluxograma



## F2

```
;Arquivo f2.pro
```

```
FUNCTION f2, x

y = x^0
A = x ## transpose(y)
B = y ## transpose(y)

RETURN, A/B
```

## F3

```
;Arquivo f3.pro
```

```
FUNCTION f3, x
y = x GT 0
A = y ## transpose(y^0)
RETURN, A
```

# F4

```
;Arquivo f4.pro
```

FUNCTION f4,  $\times$ RETURN, f3(- $\times$ )

```
; Arquivo s2.pro
PRO s2, a, b
   M = [[a], [b]]
   M = tranpose(M)
   OPENW, loc, 'dados.txt', /get lun
   PRINTF, loc, M
   CLOSE, loc
   FREE LUN, loc
END
```

; Utiliza modelo de Okumura Hata

#### PRO con01b

```
X = dindgen(146)/10+1.5 ; Distância da antena em km
POT = 16 ; Potencia de transmissão em dB
f = 800 ; Frequencia em MHz
hb = 100 ; Altura da antena transmissora em metros
hm = 10 ; Altura da antena receptora em metros
```

Cálculo da perda de potencial

```
Y = POT - ( 69.55 + 26.16*alog10(f) - 13.82*alog10(hb) - $
  ( 0.8 + (1.1*alog10(f) - 0.7)*hm - 1.56*alog10(f) ) + $
  (44.9 - 6.55*alog10(hb))*alog10(X) )
```

PLOT, X, Y

### S1

#### PRO s1

```
X = dindgen(146)/10+1.5

POT = 16 ;Potencia de transmissão em dB
f = 800 ;Frequencia em MHz
hb = 100 ; Altura da transmissora (m)
hm = 10 ; Altura da antena receptora (m)
Y = POT - f1(X,f,hb,hm)
s2, X, Y
```

## F<sub>1</sub>

### FUNCTION f1, x, f, hb, hm

```
A = 69.55 + 26.16*alog10(f)
B = (1.1*alog10(f) - 0.7)*hm
C = 0.8 + B - 1.56*alog10(f)
D = (44.9 -6.55*alog10(hb))*alog10(X)

Y = ( A - 13.82*alog10(hb) - C + D)
N = x^0 ## transpose(x^0)
Y = Y + 8*randomn(SEED, N)
```

RETURN, Y

## **S**3

```
PRO s3
  OPENR, loc, 'dados.txt', /get lun
  M = dindgen(2, file lines('dados.txt'))
  READF, loc, M
  COMMON, share, a, b
  a = transpose(M[0,*])
  b = transpose(M[1, *])
  CLOSE, loc
  FREE LUN, loc
END
```

## **S4**

### **PRO** s4

COMMON, share, a, b

med = f2(b)

npos = f3(b)

nneg = f4(b)

\$5, med, npos, nneg

```
PRO s5, med, npos, nneg
  COMMON, share, a, b
  OPENW, loc, 'relatorio.txt', /get lun
  PRINTF, loc, 'Mini-relatorio'
  PRINTF, loc, 'media = ', med, 'n positivos', npos, $
    'n negativos = ', nneg
  PRINTF, loc, 'coluna a e coluna b'
  PRINTF, loc, [[a],[b]]
  CLOSE, loc
  FREE LUN, loc
END
```

s6

**PRO** s6

COMMON, share, a, b
PLOT, a, b

# Script (arquivodescript.wtf)

S1

S3

S4

S6

# Dúvidas?

HTTP://IDLTUTORIAL.BLOGSPOT.COM

ANTONIOPAULOVP@GMAIL.COM

LUCIOMARASSI@GMAIL.COM