

# IDL

Interactive Data Language

# Revisão

- \* Álgebra Booleana
  - \* AND, OR, NOT, True/False, 1/0
- \* Álgebra Matricial
  - \* Vetor, Matriz, operações com matrizes, produto matricial.
- \* Fluxograma/Diagrama de Blocos

# Revisão

- \* IDL
  - \* Console, Execução, Função, Procedimento.
- \* Memória
  - \* Variáveis, Tipos, Tamanhos, Regras de nomenclatura, Declaração.
- \* PRINT, READ, DELVAR

# Revisão

- \* Operadores matemáticos

- \* +, -, \*, /, MOD, ^

- \* Comparadores

- \* GE, GT, LE, LT, EQ, NE

- \* Operadores Especiais

- \* <, >

# Revisão

- \* Operadores Lógicos
  - \* Normais: `&&`, `||`, `~`
  - \* Bit-a-bit: AND, OR, NOT, XOR
- \* Operadores incrementais
  - \* `++A`, `A++`, `--A`, `A--`
- \* Marcadores Especiais
  - \* `$`, `&`, `;`

# Revisão

- \* Constantes

- \* !RADEG, !DTOR, !VALUES.F\_INFINITY, !VALUES.F\_NAN
- \* !PI, !DPI, !VALUES.D\_INFINITY, !VALUES.D\_NAN

- \* Funções

- \* ALOG, ALOG10, EXP, SQRT, SIN, COS, TAN
- \* BEEP, WAIT, SYSTIME(0), SYSTIME(1).

# Revisão

- \* Aleatoriedade
  - \* RANDOMN(SEED), RANDOMU(SEED)
- \* Arrays
  - \* Adimensionais, Unidimensionais, Bidimensionais, Tridimensionais, Tetradimensionais, Ndimensionais.
- \* Funções geradoras
  - \* INDGEN, DINDGEN, FINDGEN, CINDGEN, LINDGEN.

# Revisão

- \* Operadores e funções vetoriais
  - \* ##, #, TRANSPOSE
- \* Procedimentos
  - \* **PRO**
  - \* ;Código
  - \* **END**



# Revisão

- \* IF...THEN...ELSE

- \* BEGIN...END

- \* GOTO

  - \* LABEL:

- \*  $X = \_? \_:$

# Revisão

- \* Estrutura de Escolha

- \* SWITCH

- \* CASE

- \* Break

- \* Funções Conversoras

- \* FIX(), DOUBLE(), COMPLEX(,), FLOAT(), LONG().

# Loop

Comeca\_Denovo:

$X = X + 1$

$Y = X^2$

PRINT, Y

IF X LT 20 THEN GOTO, Comeca\_Denovo

# Estruturas de Loops

## \* FOR

**FOR** count=0, 50 **DO** <ação>

**FOR** count=0, 50, 0.5 **DO** <ação>

## \* WHILE

**WHILE** <condição> **DO** <ação>

## \* REPEAT

**REPEAT** <ação> **UNTIL** <condição>

# Uso do BEGIN

```
FOR <>, <> DO BEGIN  
<>  
<>  
ENDFOR
```

```
WHILE <> DO BEGIN  
<>  
<>  
ENDWHILE
```

```
REPEAT BEGIN  
<>  
<>  
ENDREP UNTIL <>
```

# Comparação

- \* FOR

- \* Sempre realiza uma contagem.
- \* Finaliza quando a contagem supera o valor máximo.

- \* WHILE

- \* A condição é testada, e enquanto ela se mantiver verdadeira a estrutura se repetirá.

- \* REPEAT

- \* O código é executado até que a condição se torne verdadeira.

# Exemplo

**PRO** ex\_01

`X = dindgen(50)`

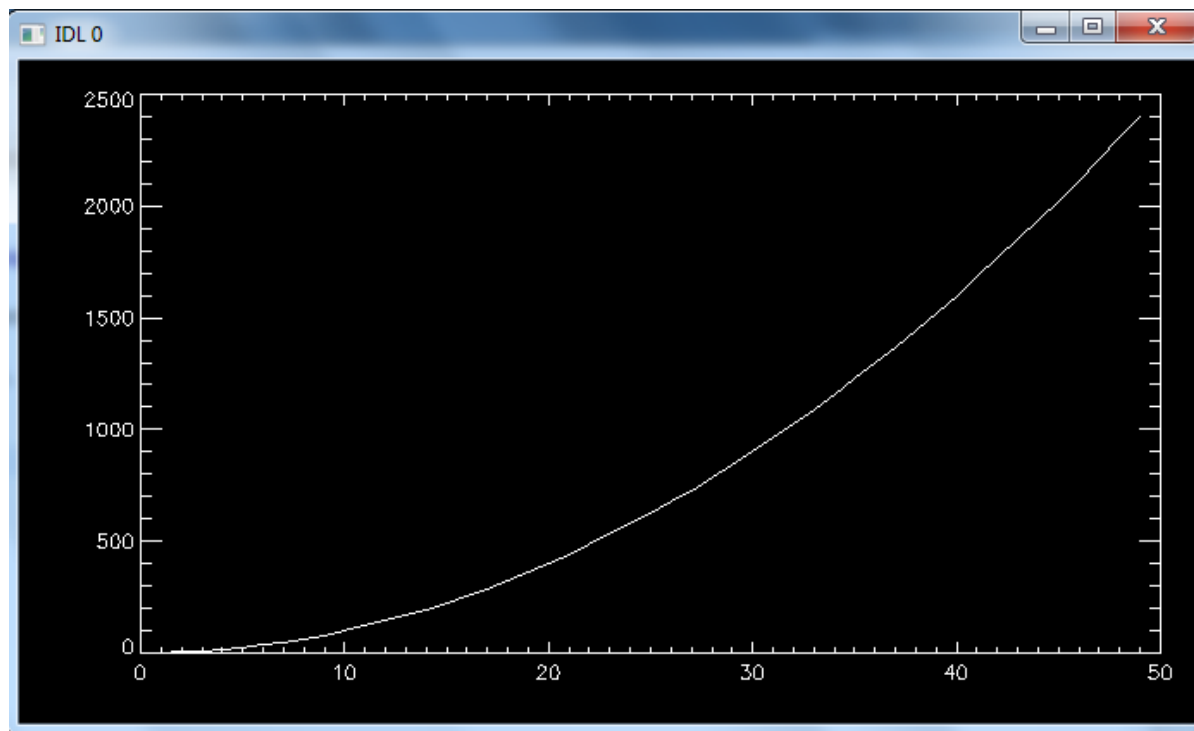
`Y = x`

`FOR count=0, 49 DO Y[count] = X[count]^2`

`PLOT, X, Y ;Plota um gráfico Y x X`

**END**

# Janela de Gráfico





# Atividade

- \* Crie um algoritmo para verificar se um número é primo ou não.
- \* Crie um algoritmo que exiba os primeiros 15 termos da série de fibonacci.
- \* Crie um algoritmo que gere 2000 números aleatórios, de distribuição gaussiana com média 10 e variância 5, e informe quantos deles eram inferiores a 0.

# Atividade

- \* Gere um vetor de números aleatórios de média 0 e variância 20, e informe quantos estão na faixa de -0.5 à 0.5.
- \* Gere um vetor de números aleatórios de média 0 e variância 50, e informe a soma dos positivos, a soma dos negativos, e o número de zeros.

# Atividades

- \* Crie um algoritmo que gere 20 números aleatórios e os ordene do menos para o maior.
- \* Crie um algoritmo que calcule a média e o desvio padrão dos termos de um vetor aleatório, de média e variâncias aleatórias.
  - \* A média e a variância são dadas por  $\text{randomn}(\text{seed}) * 10$

# Funções e Procedimentos

## Procedimentos

- \* Iniciado por **PRO**
- \* Salvo em arquivo .pro de mesmo nome.
- \* Múltiplas Entradas
- \* Múltiplas saídas
- \* Entradas e saídas se misturam
- \* Usado como linha a ser executada

## Funções

- \* Iniciado por **FUNCTION**
- \* Salvo em arquivo .pro de mesmo nome.
- \* Múltiplas entradas
- \* Uma saída
- \* Entradas e saídas não se misturam
- \* Usado como valor a ser atribuído

# Funções e Procedimentos

## Procedimento

**PRO** ex\_02a

;código

**END**

## Função

**FUNCTION** ex\_02b

;código

**END**

# Declaração de Entradas e Saídas

## Procedimentos

```
PRO ex_03a, x, y
```

```
    y = x^2
```

```
END
```

## Funções

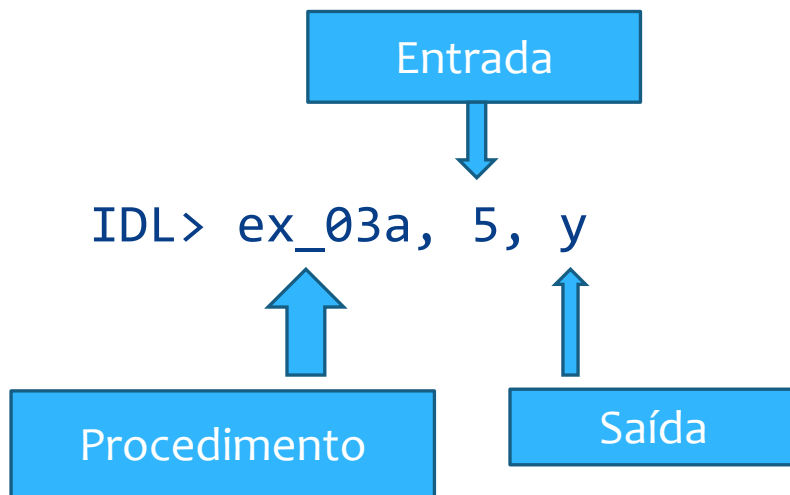
```
FUNCTION ex_03b, x
```

```
    RETURN, x^2
```

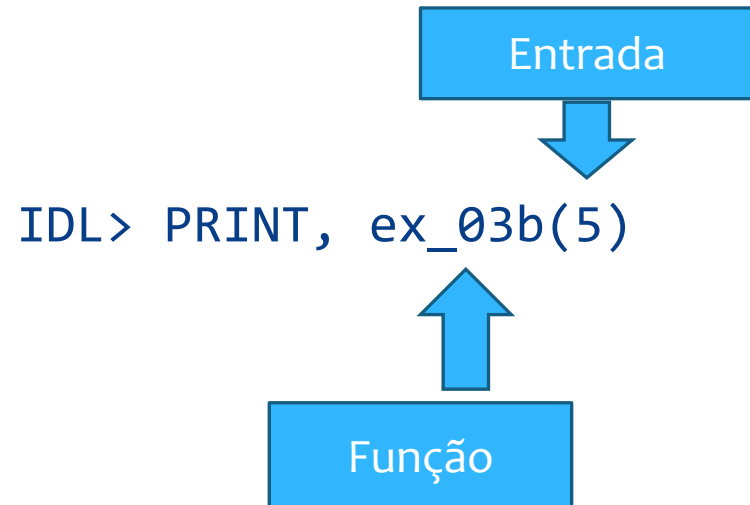
```
END
```

# Usando Procedimentos e Funções

## Procedimentos



## Funções



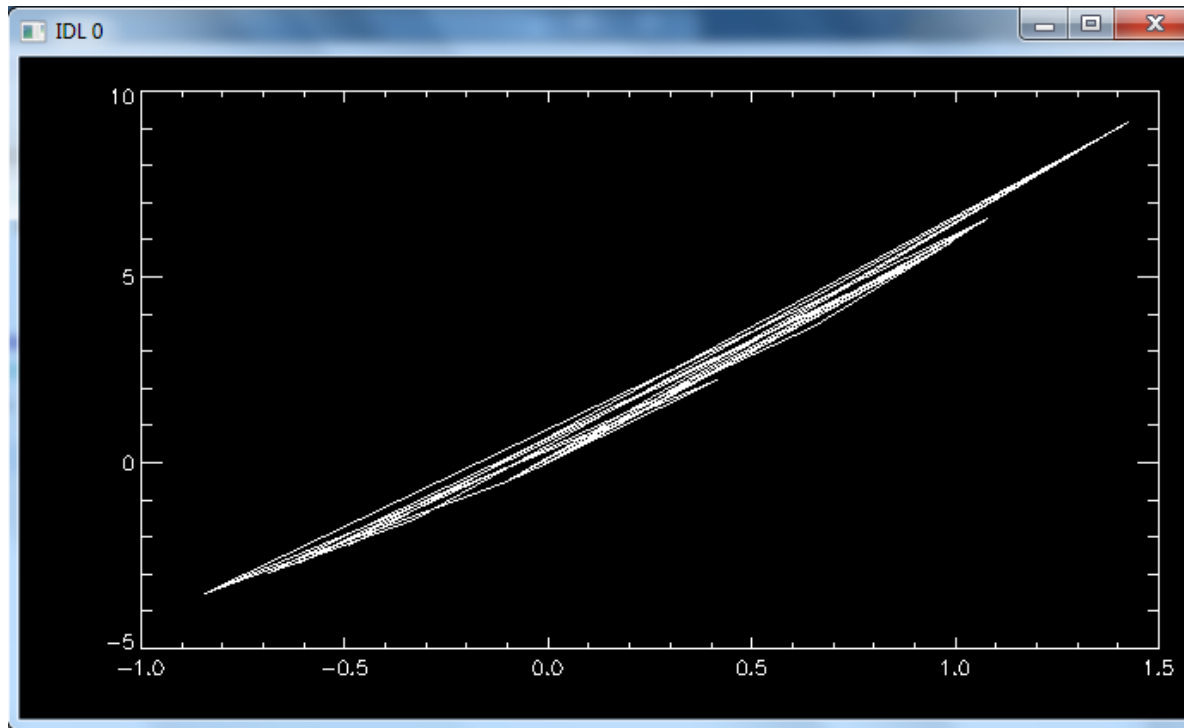
# Exemplo de Código ( ex\_04.pro )

```
FUNCTION ex_05, a  
    RETURN, a^2+5*a  
END
```

```
PRO ex_04  
    x = randomn(seed,20)  
    y = ex_05(x)  
    PLOT, x, y  
END
```



# Exemplo de Código ( ex\_04.pro )



# Scripts

- \* Conjunto de linhas de código a ser executado sequencialmente.
- \* Não importa a extensão salva.
- \* Não tem estrutura de declaração.
- \* Todas as variáveis são globais.
- \* Não tem entradas nem saídas.
- \* É executada por meio do comando:
  - \* IDL> @nome\_do\_script.ext

# Diagrama de blocos

- \* Identificar que variável vem de que função e suas dependências.
- \* Identificar as entradas e saídas de cada função.
- \* Organizar e subdividir o trabalho.
- \* Facilita o teste das partes do algoritmo, facilita no reparo de erros.

# Dúvidas?

E-mail: [antoniopaulovp@gmail.com](mailto:antoniopaulovp@gmail.com)

Blog: <http://idltutorial.blogspot.com>