

## Universidad Nacional de Trujillo

VIII Fast WorkShop on Applied and Computational Mathematics

Trujillo, Perú.

January 7-8, 2015

Programming Embedded Components for Graphics in Maple

Lenin Araujo Castillo  
physicsleninac@hotmail.com

Escuela de Postgrado  
Universidad Nacional de Trujillo

08 de Enero, 2015

# Contenido

- 1 **Introducción**
  - Programming Graphics in Maple
- 2 **Principio de Funcionamiento**
  - Para lograr la interactividad
- 3 **Maple en programación**
  - Procedure con Components Embedded
  - Casos en Programación
- 4 **Visualización del código**
  - Integral e iteración

# Desarrollo del Contenido

- 1 **Introducción**
  - Programming Graphics in Maple
- 2 Principio de Funcionamiento
  - Para lograr la interactividad
- 3 Maple en programación
  - Procedure con Components Embedded
  - Casos en Programación
- 4 Visualización del código
  - Integral e iteración

# Lenguaje de Programación

## proc() y componentes embeded

Maple a desarrollado un forma de programar mediante dos tipos (utilizando proc() y components embeded) los cuales involucran cosas aparentemente diferentes. Aqui observaremos que se puede unir la parte procedimental con la interactividad para de esa manera ser cargado en cualquier dispositivo móvil. Además; resultado de ésto y mediante el uso de otros productos podemos hacer que se comuniquen entre paquetes.



Para lograr la interactividad

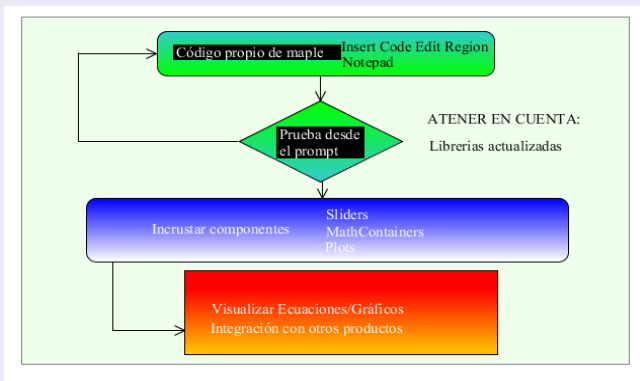
## Desarrollo del Contenido

- 1 Introducción
  - Programming Graphics in Maple
- 2 Principio de Funcionamiento
  - Para lograr la interactividad
- 3 Maple en programación
  - Procedure con Components Embedded
  - Casos en Programación
- 4 Visualización del código
  - Integral e iteración

Para lograr la interactividad

## Con Maple

Para diseñar y resolver problemas/cálculo



# Desarrollo del Contenido

- 1 Introducción
  - Programming Graphics in Maple
- 2 Principio de Funcionamiento
  - Para lograr la interactividad
- 3 Maple en programación
  - Procedure con Components Embedded
  - Casos en Programación
- 4 Visualización del código
  - Integral e iteración

# Programación con Procedimiento

## Definición

Un procedimiento de Maple; es una secuencia de declaraciones de parámetros, declaraciones de variables y declaraciones que encapsula un cálculo. Una vez definido, un procedimiento puede ser utilizado para realizar el mismo cálculo en varias ocasiones por diferentes valores de los argumentos, desde diferentes lugares del programa, o ambos.

```
proc( parameterDeclarations ) :: returnType;
    description shortDescription;
    option optionSequence;
    local localVariableDeclarations;
    global globalVariableDeclarations;
    statementSequence
end proc
```



# Programación con elementos interactivos

## Definición

La interfaz estándar de Maple que ofrece varias herramientas para la creación de contenido matemático interactivo. Incluye un conjunto de componentes integrados, que son configurables controles gráficos, botones, medidores y otros componentes interactivos que se pueden insertar en un documento de Maple para analizar, manipular y visualizar ecuaciones y comandos de Maple.



# Desarrollo del Contenido

- 1 Introducción
  - Programming Graphics in Maple
- 2 Principio de Funcionamiento
  - Para lograr la interactividad
- 3 Maple en programación
  - Procedure con Components Embedded
  - Casos en Programación
- 4 Visualización del código
  - Integral e iteración

## Ejercicio 1

## Suma



(\*Suma de N términos utilizando proc()\*)



Coimprobando con la sintaxis en el prompt:

```
sum(i, i = 0 ..100);
```

5050

## Ejercicio 2

## Derivada



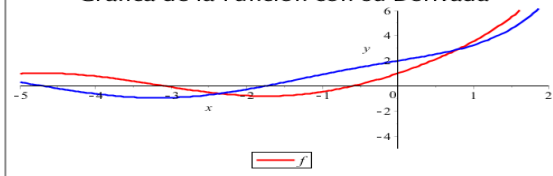
```
g_func_deriv:=proc(f1, rx::name=range, ry::name=range)
```

x0 =  xf =

y0 =  yf =



## Gráfica de la Función con su Derivada



## Ejercicio 3

## Integral



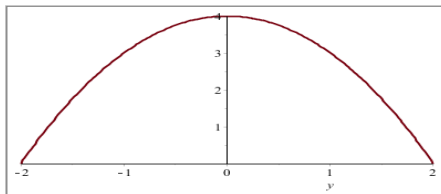
```
int_sim:=proc(f,r::name=range)
```

$y_0 =$    $y_f =$

$$4 - y^2$$

Calcular y Graficar

$$\int_{-2}^2 (-y^2 + 4) dy = 10.667$$



Comprobando en el promt

```
int(4 - y^2, y = -2 .. 2) = evalf(int(4 - y^2, y = -2 .. 2), 5);
```

$$\int_{-2}^2 (-y^2 + 4) dy = 10.667$$

## Ejercicio 4

## Iteración

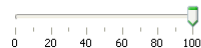
Consideremos la siguiente situación. La población de una especie que aumenta cada año en un 10%. Dejando que  $P_n$  sea la población al final de  $n$  años y  $P_0 = 100$  sea la población inicial, esta situación es modelada por la ecuación de diferencias (o relación de recurrencia):

$$P_{k+1} = 1.1P_k, \quad P_0 = 100, \quad k=0, 1, \dots, n$$



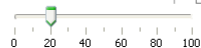
Poblacion:=proc(P0, n, A0)

$P_0 =$



100

$n =$

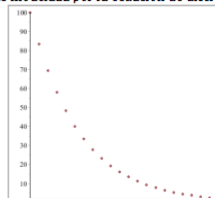


20

$A_0 =$



0.83333333



```
[0, 100], [1, 83.33333300], [2, 69.44444389], [3, 57.87036968], [4, 48.22530787], [5, 40.18775640], [6, 33.48979687], [7, 27.90816395], [8, 23.25680320], [9, 19.38066926], [10, 16.15055765], [11, 13.45879799], [12, 11.21566495], [13, 9.346387421], [14, 7.788656153], [15, 6.490546768], [16, 5.408788952], [17, 4.507324109], [18, 3.756103409], [19, 3.130086162], [20, 2.608405125]
```

Visualizar



# Desarrollo del Contenido

- 1 Introducción
  - Programming Graphics in Maple
  
- 2 Principio de Funcionamiento
  - Para lograr la interactividad
  
- 3 Maple en programación
  - Procedure con Components Embedded
  - Casos en Programación
  
- 4 Visualización del código
  - Integral e iteración

# Union de proc() y components embedded

## Ejemplo 3

The image shows two parts of a Maple environment. The top part is a code editor window containing the following Maple procedure:

```

1 int_sim:=proc(f,r::name=range)
2     local I1, vn,y;
3     y:=args[1];
4     I1:=Int(f,r);
5     vn:=simplify(value(I1), symbolic);
6     print("La Integral simple es:", I1 = vn);
7     print(I=evalf(vn));
8     plot(y, r, thickness=2);
9 end proc;

```

A green arrow points from a gear icon to the procedure code. Below the code editor is a window titled "Button2 Action When Clicked" showing the following Maple code:

```

1 use DocumentTools in
2 Do(%Plot1:=int_sim(%MathContainer7,y=%Slider5..%Slider6));
3 Do(%MathContainer8=Int(%MathContainer7,y=%Slider5..%Slider6));
4 Do(%MathContainer9=evalf(int(%MathContainer7,y=%Slider5..%Slider6),5));
5 end use;

```

A green arrow points from a button labeled "Calcular y Graficar" to the second line of code in the button action window. The bottom of the button action window shows "No errors".



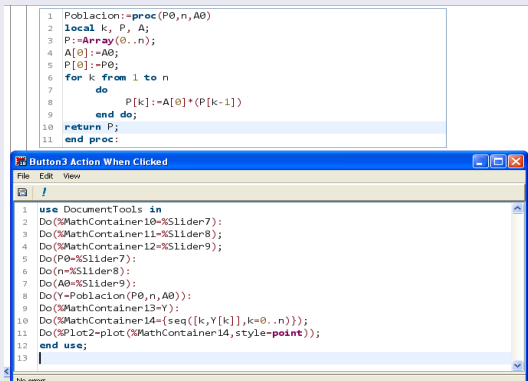
# Union de proc() y components embedded

## Ejemplo 4

```

1 Poblacion:=proc(P0,n,A0)
2 local k, P, A;
3 P:=Array(0..n);
4 A[0]:=A0;
5 P[0]:=P0;
6 for k from 1 to n
7 do
8     P[k]:=A[0]*(P[k-1])
9 end do;
10 return P;
11 end proc;

```



```

1 use DocumentTools in
2 Do(%MathContainer10=%Slider7);
3 Do(%MathContainer11=%Slider8);
4 Do(%MathContainer12=%Slider9);
5 Do(P0=%Slider7);
6 Do(n=%Slider8);
7 Do(A0=%Slider9);
8 Do(Y:=Poblacion(P0,n,A0));
9 Do(%MathContainer13=Y);
10 Do(%MathContainer14={seq([k,Y[k]],k=0..n)});
11 Do(%Plot2=plot(%MathContainer14,style=point));
12 end use;
13




```

No errors

## Conclusiones y trabajos futuros

- La interacción `proc()` con `components embedded`.
- Tiempo de procesamineto mínimo.
- Innovación en construcciones de modelos matemáticas.
- A mediano plazo.
  - Compartir ésta información a diferentes especialidades.
  - Cursos y Talleres de casos aplicados a la enseñanza y la ingeniería.
  - Próximamente: programación con sistema táctil.

# Lecturas adicionales I

-  JTHOMAS WESTERMANN, *Mathematische Probleme lösen mit Maple*, © Springer-Verlag Berlin Heidelberg, 2014,
-  MAZEN SHAHIN, *Explorations of Mathematical models in Biology with Maple*, JohnWiley Dover 2015
-  WILHELM FORST, *Funktionentheorie erkunden mit Maple*, Springer 2012

# Gracias por su participación

¿PREGUNTAS?

Muchas Gracias!!!

Para saber más.

[www.maplesoft.com](http://www.maplesoft.com)

[www.mapleprimes.com](http://www.mapleprimes.com)