



Conceptos de Algoritmos Datos y Programas

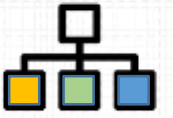
CADP – TEMAS



- Modularización - FUNCIONES
- EJERCICIO – PREGUNTAS FINALES

CADP – MODULARIZACION

FUNCIONES



Programa nombre

areas

...

Procesos

proceso nombre (parámetros)

variables

comenzar

fin

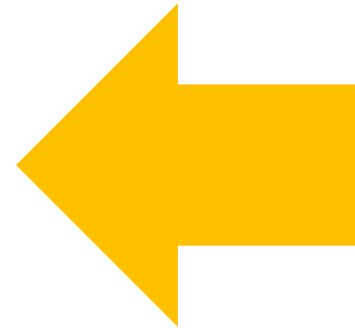
Variables

...

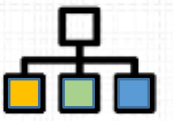
comenzar

...

fin



Cómo son?
Cómo se declaran?
Cómo se usan?



FUNCION

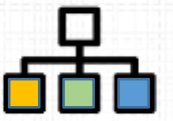
Conjunto de instrucciones que realizan una tarea especifica y retorna 1 valor de tipo simple.

```
function nombre (parámetros): tipo;  
var  
    ....  
begin  
    ....  
end;
```

} Variables locales

} Código de la función

CADP – MODULARIZACION FUNCIONES



```
Program uno;  
Const  
    ....  
Type  
    ....  
function auxiliar (parametros): tipo;  
Var  
    x:integer;  
begin  
    x:=8;  
    ...  
    auxiliar:= valor que se quiere retornar;  
end;  
Var  
    ....  
Begin  
    ...  
End.
```

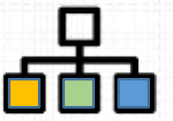
La función auxiliar no tiene parámetros

La función auxiliar tiene una variable x local declarada

Tipo debe ser un tipo de datos simple

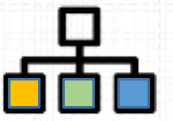
Para retornar el valor la última instrucción de la función debe ser asignarle a su nombre el valor que se quiere retornar

CADP – MODULARIZACION FUNCIONES



```
Program uno;  
Const  
    ....  
Type  
    ....  
function auxiliar:integer;  
Var  
    x,resto:integer;  
begin  
    x:=8;  
    resto:= x MOD 5;  
    ...  
    auxiliar:= resto;  
end;  
Var  
    ....  
Begin  
    ...  
End.
```

Cómo se invocan?



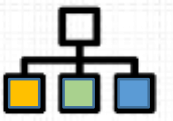
INVOCACION POR SU NOMBRE

Invocación usando variable

El resultado se asigna
a una variable del
mismo tipo que
devuelve la función.

```
program uno;  
Function auxiliar: real;  
Var  
    x, y, cociente:real;  
  
begin  
    x:= 10;  
    y:= 4;  
    cociente:= x/y;  
    auxiliar:= cociente;  
end;  
Var  
    aux:real;  
begin  
    aux:= auxiliar;  
    write (aux);  
end.
```

El retorno de la
función es a la
misma línea de
invocación



INVOCACION POR SU NOMBRE

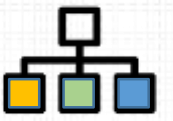
Invocación en un while
o
en un if

El resultado se utiliza
para evaluar la
condición.

```
program uno;  
Function auxiliar: real;  
Var  
    x, y, cociente:real;  
  
begin  
    x:= 10;  
    y:= 4;  
    cociente:= x/y;  
    auxiliar:= cociente;  
end;
```

```
begin  
    while (auxiliar = 5.5) do  
  
        if (auxiliar = 5.5) then  
  
end.
```

El retorno de la
función es a la misma
línea de invocación



INVOCACION POR SU NOMBRE

Invocación en un write

El resultado se utiliza
para informar en la
sentencia write.

```
program uno;
```

```
Function auxiliar: real;
```

```
Var
```

```
    x, y, cociente:real;
```

```
begin
```

```
    x:= 10;
```

```
    y:= 4;
```

```
    cociente:= x/y;
```

```
    auxiliar:= cociente;
```

```
end;
```

```
begin
```

```
    write ('El resultados es,auxiliar);
```

```
end.
```