

## CONCEPTOS Y VENTAJAS DE LA MODULARIZACIÓN DE PROGRAMAS

### ENUNCIADO

Una empresa de servicios industriales requiere un programa de análisis estadístico de datos, que le permita calcular la media aritmética y la desviación estándar de elementos que deberán almacenarse en un arreglo unidimensional.

### ANÁLISIS

El problema presenta tres requisitos esenciales:

- cargar los datos en un vector,
- calcular la media aritmética,
- calcular la desviación estándar de los datos.

Podemos ver que tenemos un problema que puede ser desglosado en tres subproblemas más sencillos de resolver.

Una de las ventajas de la Modularización es precisamente el “divide y vencerás”, es decir que la resolución de un problema se torna más simple a medida que lo descomponemos en subproblemas de menor complejidad individual. Otra de las ventajas es la reusabilidad del código fuente, pero escapa del alcance de la asignatura.

Veamos las dos formas de resolver esto mismo: sin modularizar y con modularización.

### SIN MODULARIZAR

```
program media_aritmetica;
uses crt;
const
    DIM=30; {Número de elementos}

Type
    Vec = array [1..DIM] of real;

Var
    a:array;
    n:integer;
    i,suma,media,d:real;
begin
    repeat
        write ('Introduzca el número de elementos ');
        readln (n);
    until (n>=DIM);
    suma:=0;
    for i:=1 to n do
    begin
        readln (a[i]);
        suma:=suma+a[i];
    end;
    media:=suma/DIM;
    writeln ('La media aritmética vale ',media:6:2);
    writeln;
    d:=0;
    for i:=1 to n do
        d:=d+((a[i]-media)*(a[i]-media));
    writeln(La desv. estándar vale ,sqrt(d/n):6:2);
end.
```

**CON MODULARIZACIÓN**

```
program media_aritmetica;
const
  DIM=30; {Número de elementos}
type
  MiArray = array[1..DIM] of real;
var
  a: MiArray;
  media,d:real;
  n:integer;

procedure LeeDatos(VAR elem: MiArray; VAR num:integer);
var
  i:integer;
begin
  repeat
    write ('Introduzca el número de elementos ');
    readln (num);
  until (num<=DIM);
  for i:=1 to num do
  begin
    write ('Introduzca el dato ', i, ' : ');
    readln (elem[i]);
  end;
end;

function CalculaMedia(VAR elem: MiArray; num:integer): real;
var
  suma:real;
  i:integer;
begin
  suma:=0;
  for i:=1 to num do
  begin
    suma:=suma+elem[i];
  end;
  CalculaMedia:=suma/n;
end;

function CalculaDesviacion(VAR elem: MiArray; media:real; num:integer):real;
var
  d:real;
  i:integer;
begin
  d:=0;
  for i:=1 to num do
    d:=d+((elem[i]-media)*(elem[i]-media));
  CalculaDesviacion:=sqrt(d/num);
end;

Begin
  LeeDatos (a,n);
  media:=CalculaMedia (a,n);
  d:=CalculaDesviacion (a,media,n);
  writeln ('La media aritmética vale ',media:6:2);
  writeln ('La desv. estándar vale',d:6:2);
end.
```

Podemos apreciar que los módulos o subprogramas se comunican con el programa principal mediante el pasaje de parámetros. Una correcta modularización simplifica la resolución de problemas y permite diseñar una estrategia de resolución más eficiente.