



# Conceptos de Algoritmos Datos y Programas



# CADP – TEMAS



- Estructura de datos ARREGLO
- Recorridos totales
- Recorridos parciales



### RECORRIDOS

Consiste en recorrer el vector de manera total o parcial, para realizar algún proceso sobre sus elementos.

#### RECORRIDO - TOTAL

Implica analizar todos los elementos del vector, lo que lleva a recorrer completamente la estructura.

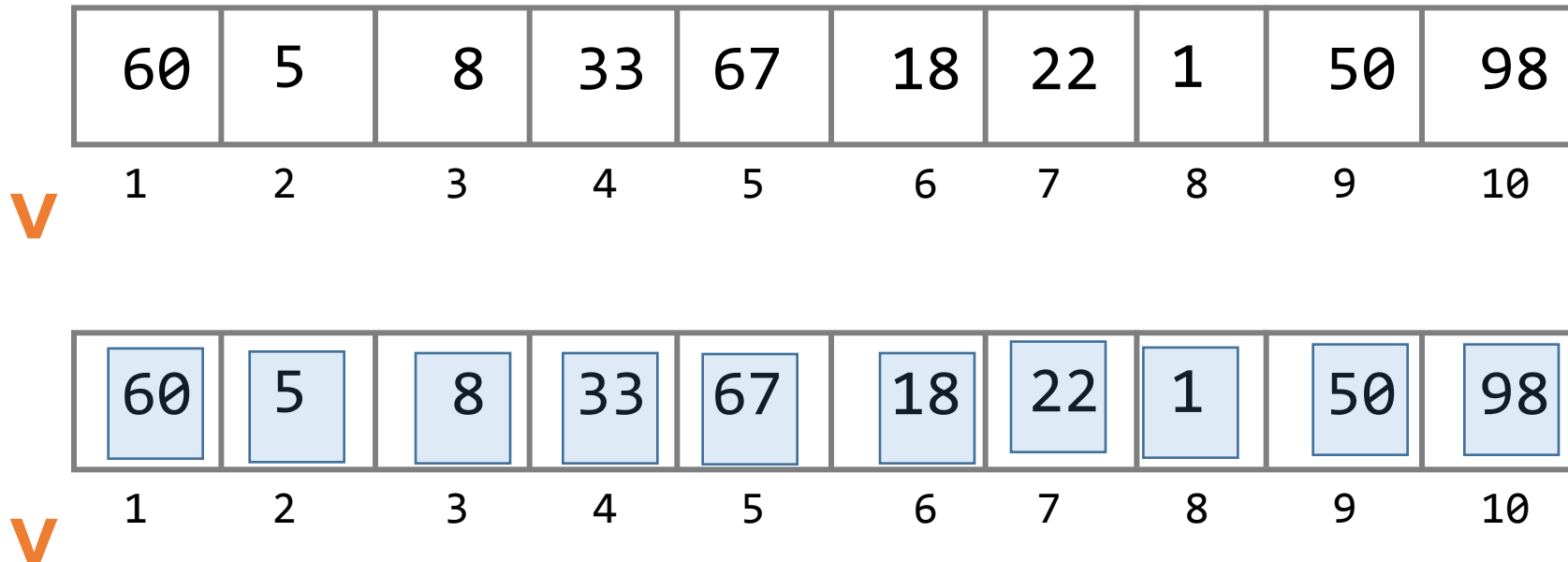
Qué estructura de control implica cada uno?

#### RECORRIDO - PARCIAL

Implica analizar los elementos del vector, hasta encontrar aquel que cumple con lo pedido. Puede ocurrir que se recorra todo el vector



Realice un programa que llene un vector de 10 elementos enteros positivos y luego informe la cantidad de números múltiplos de 3. Suponga que los nros leídos son positivos.



Cant 0  
 Cant 1  
 Cant 2  
 Cant 3

Qué estructura  
de control?

Qué  
modularizo?

Cómo lo  
implemento?

```
Program uno;
```

```
Const
```

```
  tam=10;
```

```
  multi=3;
```

```
Type
```

```
  vector = array [1..tam] of integer;
```

```
Var
```

```
  v1:vector;
```

```
  cant:integer;
```

```
Begin
```

```
  cargar (v1);
```

```
  cant:= múltiplos (v1);
```

```
  write (“La cantidad de múltiplos de”, multi, “es”, cant);
```

```
End.
```

```
Procedure cargar (var v:vector);
```

```
Var
```

```
    i,valor:integer;
```

```
Begin
```

```
    for i:= 1 to tam do
```

```
        begin
```

```
            read(valor);
```

```
            v[i]:= valor;
```

```
        end;
```

```
End;
```

## ALTERNATIVA

```
Procedure cargar (var v:vector);
```

```
Var
```

```
    i:integer;
```

```
Begin
```

```
    for i:= 1 to tam do
```

```
        begin
```

```
            read(v[i]);
```

```
        end;
```

```
End;
```

```
function multiplos (v:vector):integer;
```

Var

```
  i,cant,resto: integer;
```

Begin

```
  cant:=0;
```

```
  for i:= 1 to tam do
```

```
    begin
```

```
      resto:= V[i] MOD multi;
```

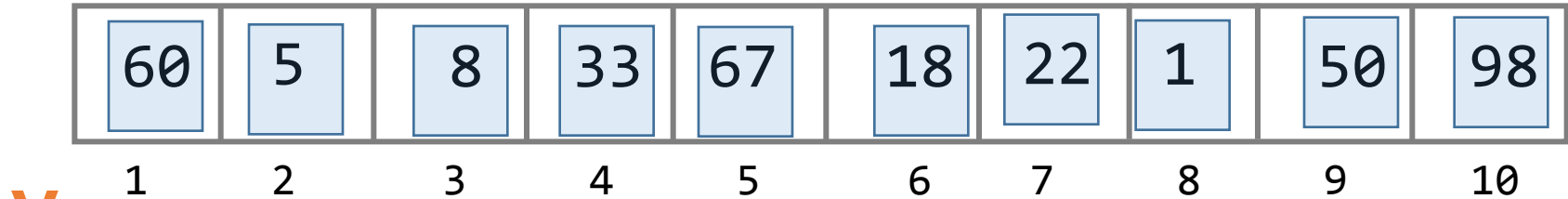
```
      if (resto = 0) then
```

```
        cant:= cant + 1;
```

```
      end;
```

```
  multiplos:= cant;
```

```
End;
```



ALTERNATIVA

```
function multiplos (v:vector):integer;
```

Var

```
  i,cant: integer;
```

Begin

```
  cant:=0;
```

```
  for i:= 1 to tam do
```

```
    begin
```

```
      if ((v[i] MOD multi) = 0) then
```

```
        cant:= cant + 1;
```

```
      end;
```

```
  multiplos:= cant;
```

```
End;
```



Realice un programa que cargue un vector de 10 elementos enteros positivos y luego informe la primer posición donde aparece un múltiplo de 3. Suponga que los nros leídos son positivos y que existe al menos un múltiplo de 3.

61	5	8	33	67	18	22	1	50	98
1	2	3	4	5	6	7	8	9	10

V



POS 4

61	5	8	33	67	18	22	1	50	98
1	2	3	4	5	6	7	8	9	10

V

Qué estructura  
de control?

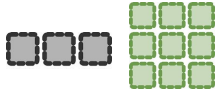
Qué  
modularizo?

Cómo lo  
implemento?



# CADP – VECTOR

## RECORRIDO - PARCIAL



```
Program uno;
```

```
Const
```

```
  tam=10;
```

```
  multi=3;
```

```
Type
```

```
  vector = array [1..tam]    of    integer;
```

```
Var
```

```
  v:vector;
```

```
  pos:integer;
```

```
Begin
```

```
  cargar (v);
```

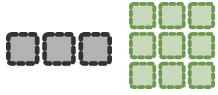
```
  pos:= posicion (v);
```

```
  write (“La posición del primer múltiplo de”, multi, “es”, pos);
```

```
End.
```

# CADP – VECTOR

## RECORRIDO PARCIAL



```
function posicion (v: vector): integer;
```

```
var
```

```
    pos, resto: integer;
```

```
    seguir: boolean;
```

61	5	8	33	67	18	22	1	50	98
1	2	3	4	5	6	7	8	9	10

V

```
begin
```

```
    seguir := true; pos := 1;
```

```
    while (seguir = true) do
```

```
        begin
```

```
            resto := v[pos] MOD multi;
```

```
            if (resto = 0) then
```

```
                seguir := false
```

```
            else
```

```
                pos := pos + 1;
```

```
            end;
```

```
        posicion := pos;
```

```
    end;
```

Por qué se  
inicializa pos en 1?

Por qué pos se  
incrementa en el else?

Qué cambio si el  
enunciado no asegura  
que haya al menos un  
múltiplo de 3?

# CADP – VECTOR

## RECORRIDO PARCIAL



```
function posicion (v: vector): integer;
```

```
var
```

```
    pos,resto:integer;
```

```
    seguir:boolean;
```

**v**

61	5	8	33	67	18	22	1	50	98
1	2	3	4	5	6	7	8	9	10

```
begin
```

```
    seguir:= true; pos:=1;
```

```
    while ((pos<= tam) and (seguir = true)) do
```

```
        begin
```

```
            resto:= v[pos] MOD multi;
```

```
            if (resto = 0) then
```

```
                seguir:= false
```

```
            else
```

```
                pos:= pos + 1;
```

```
            end;
```

```
            if (seguir = false) then posicion:= pos
```

```
                else posicion:= -1;
```

**v**

61	5	8	31	67	19	22	1	50	98
1	2	3	4	5	6	7	8	9	10

Es necesario la  
última condición  
del if?