

# Cálculo Numérico (M107)

Profesor: Nicolás G Tripp  
[ntripp@fcen.uncu.edu.ar](mailto:ntripp@fcen.uncu.edu.ar)

Aula Virtual  
<http://fcen.uncuyo.edu.ar/calculo-numerico>

# Unidad 1: Introducción a la programación científica y al cálculo numérico.

## Temario

- Algoritmos y diagramas de flujo.
- Programas secuenciales, control de flujo, repetición.
- Operaciones vectoriales y matriciales.

# Algoritmos

La palabra *algoritmo* deriva del nombre de un matemático árabe del siglo IX, llamado “Al-Khuwarizmi”, quien describió varios métodos para resolver problemas de aritmética. Un *algoritmo* es *un conjunto de instrucciones claras sobre un proceso para hacer algo*, es decir, *una receta*.



Propiedades:

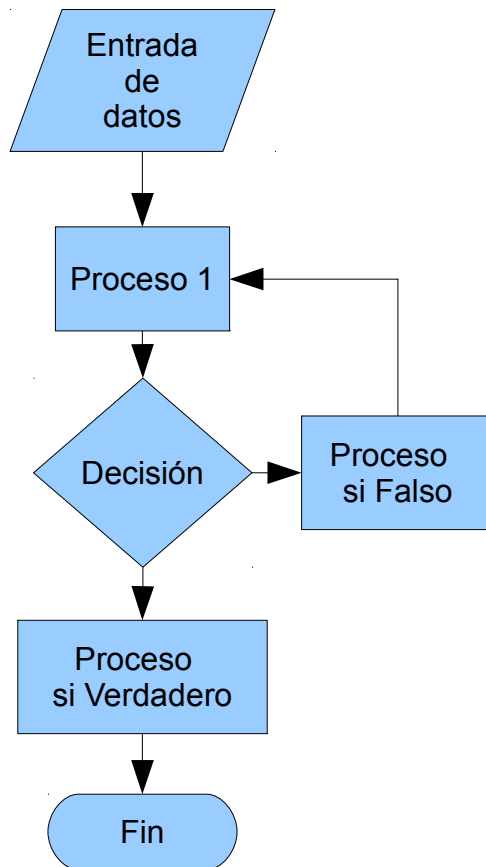
- 1) No debe ser ambiguo.
- 2) Debe detenerse.
- 3) Debe entregar un resultado.

*Antes de intentar programar la computadora,  
se debe tener claro que se quiere hacer!!!*

Para más información consultar  
 Curso de Ingreso 2014, Facultad de Informática,  
 Universidad Nacional de La Plata

# Formas de expresar un algoritmo

## Diagrama de flujo



## Pseudocódigo:

- 1) Leer datos
- 2) Ejecutar Proceso 1
- 3) Evaluación lógica
- 4a) Si es falso realizar F y volver a Proceso 1
- 4b) Si es verdadero realizar V y seguir a Fin.
- 5) Fin

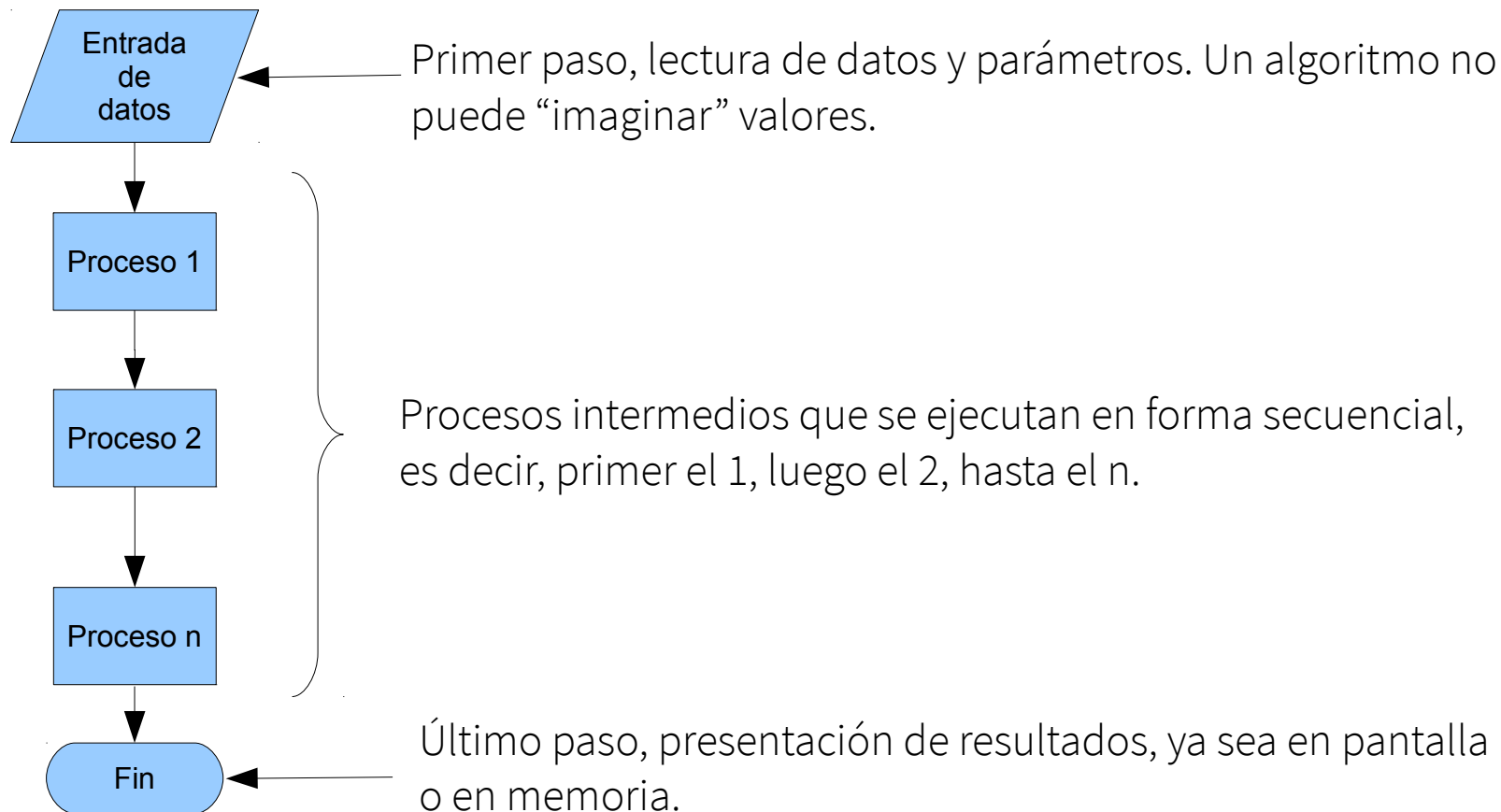
## Código fuente

```

Function test()
input=csvread('datos')
[var1]=P1(input);
while cond == False
    PF
endwhile
PV
endfunction
  
```

# Elementos básicos de diagramas de flujo

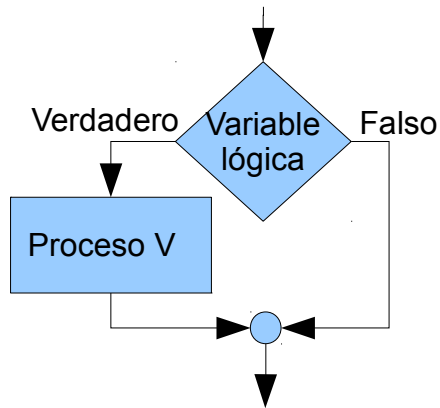
## Estructura secuencial



# Controles de flujo

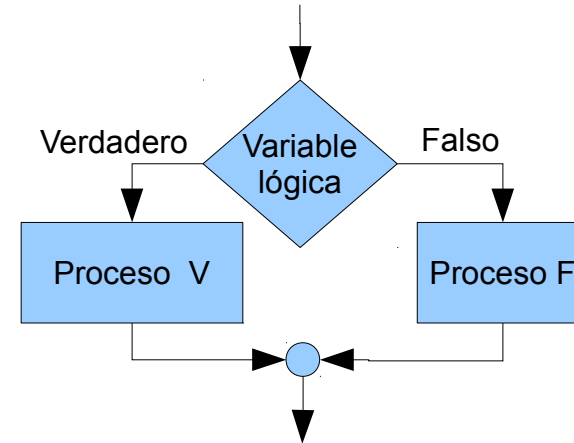
## Simple

Si la condición es Verdadera ejecuta proceso V, si es falsa no hace nada



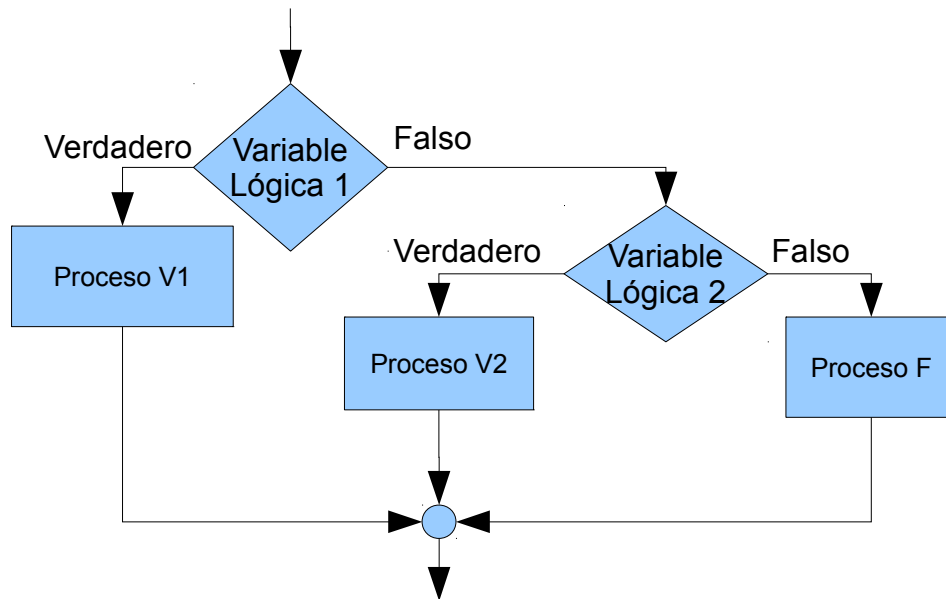
## Doble

Si la condición es Verdadera ejecuta proceso V, si es falsa ejecuta proceso F



## Anidado

Si la condición 1 es verdadera ejecuta proceso V1, si es falsa se observa la condición 2, si es verdadera se ejecuta el proceso V2 y de ser falsa se ejecuta el proceso F.

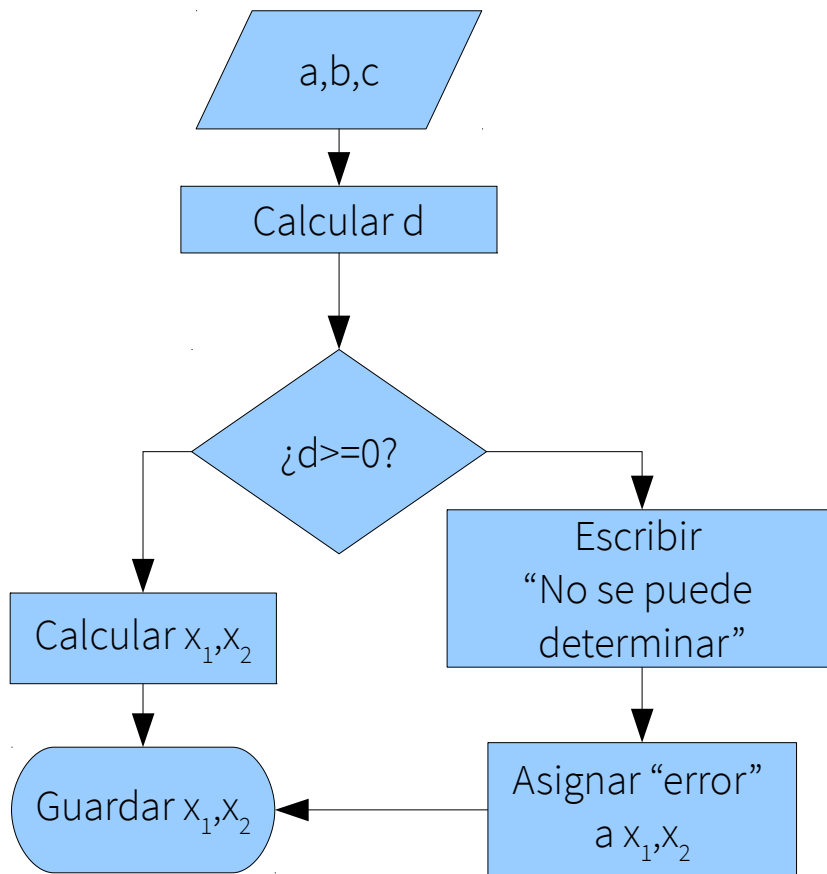


Cond 1	Cond 2	Proceso
V	V	V1
V	F	V1
F	V	V2
F	F	F

# Controles de flujo

Ejemplo, fórmula de búsqueda de raíces de una ecuación cuadrática

$$x_{1,2} = \frac{-b \pm \sqrt{d}}{2a}, d = b^2 - 4ac$$

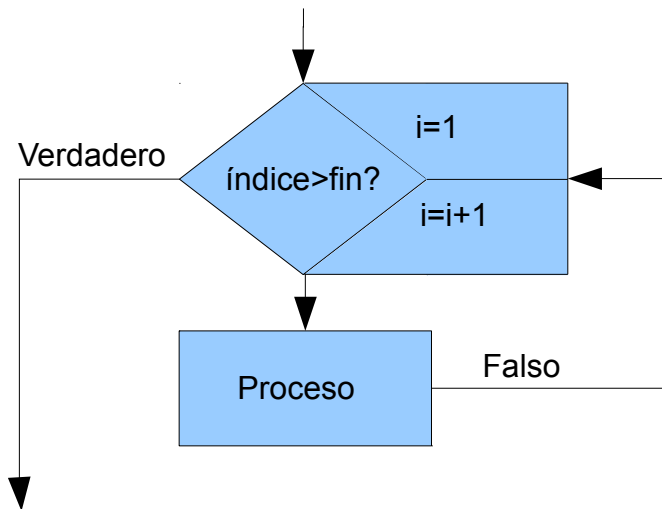


```

function [x1,x2]=solcuad()
%Solucion de ecuacion cuadratica
a=input("ingrese coeficiente a");
b=input("ingrese coeficiente b");
c=input("ingrese coeficiente c");
d=b^2-4*a*c;
if d>=0
    sol1=(-b-sqrt(d))/(2*a);
    sol2=(-b+sqrt(d))/(2*a);
else
    disp("No se puede determinar");
    sol1="error";
    sol2="error";
endif
x1=sol1;
x2=sol2;
endfunction
  
```

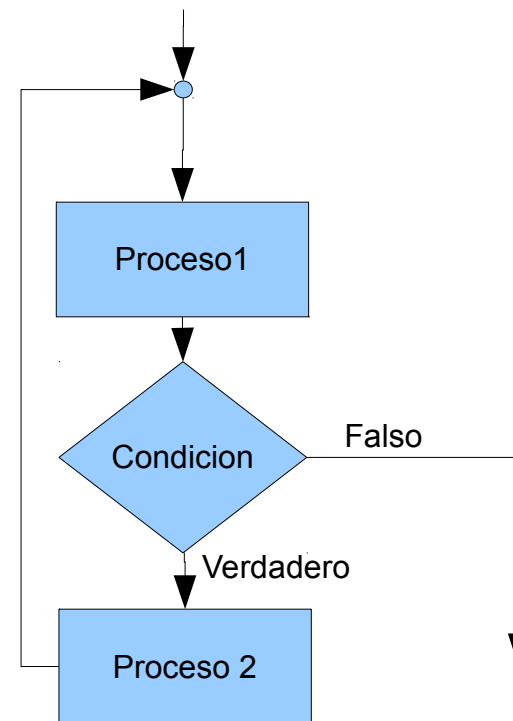
# Repeticiones de procesos

## Repetición



La repetición (“FOR”) es la estructura de control que permite al algoritmo ejecutar un conjunto de instrucciones un número de veces fijo y conocido de antemano

## Iteración

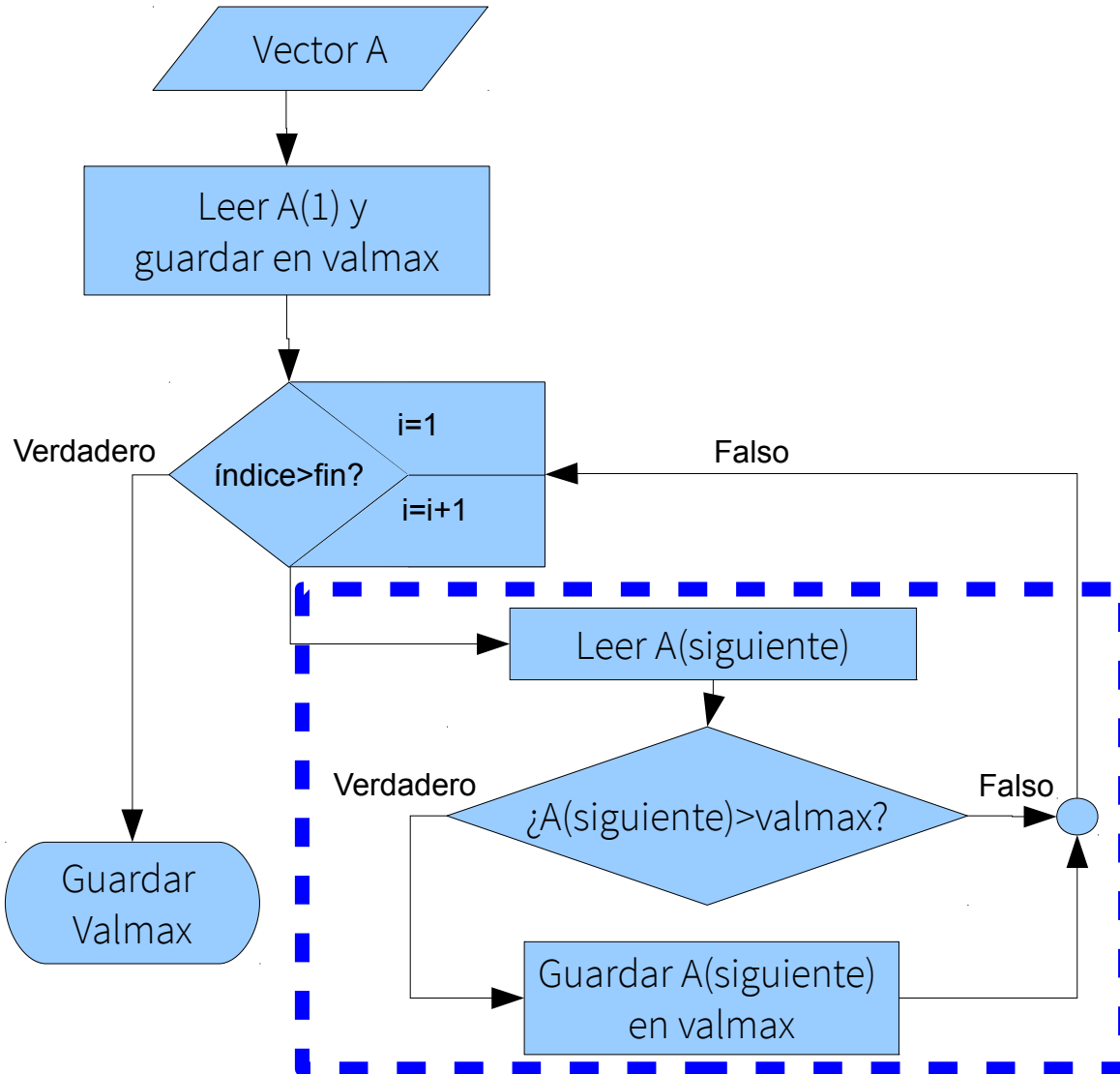


La iteración (“WHILE”) es una estructura de control que permite al algoritmo ejecutar en forma repetitiva un conjunto de acciones mientras se mantenga una condición.



# Operaciones sobre vectores

## Búsqueda de valor máximo



```
function [valmax]=buscarmax(A)
    %Buscar valor máximo del vector A
    valmax=A(1);
    N=length(A); %ver tamaño del vector
    for i=2:N
        if A(i)>valmax
            valmax=A(i);
        endif
    endfor
endfunction
```

# Operaciones sobre vectores

## Búsqueda de valor mínimo

```
function [valmin]=buscarmin(A)
    %Buscar valor mínimo del vector A
    valmin=A(1);
    N=length(A); %ver tamaño del vector
    for i=2:N
        if A(i)<valmix
            valmin=A(i);
        endif
    endfor
endfunction
```

# Operaciones sobre vectores

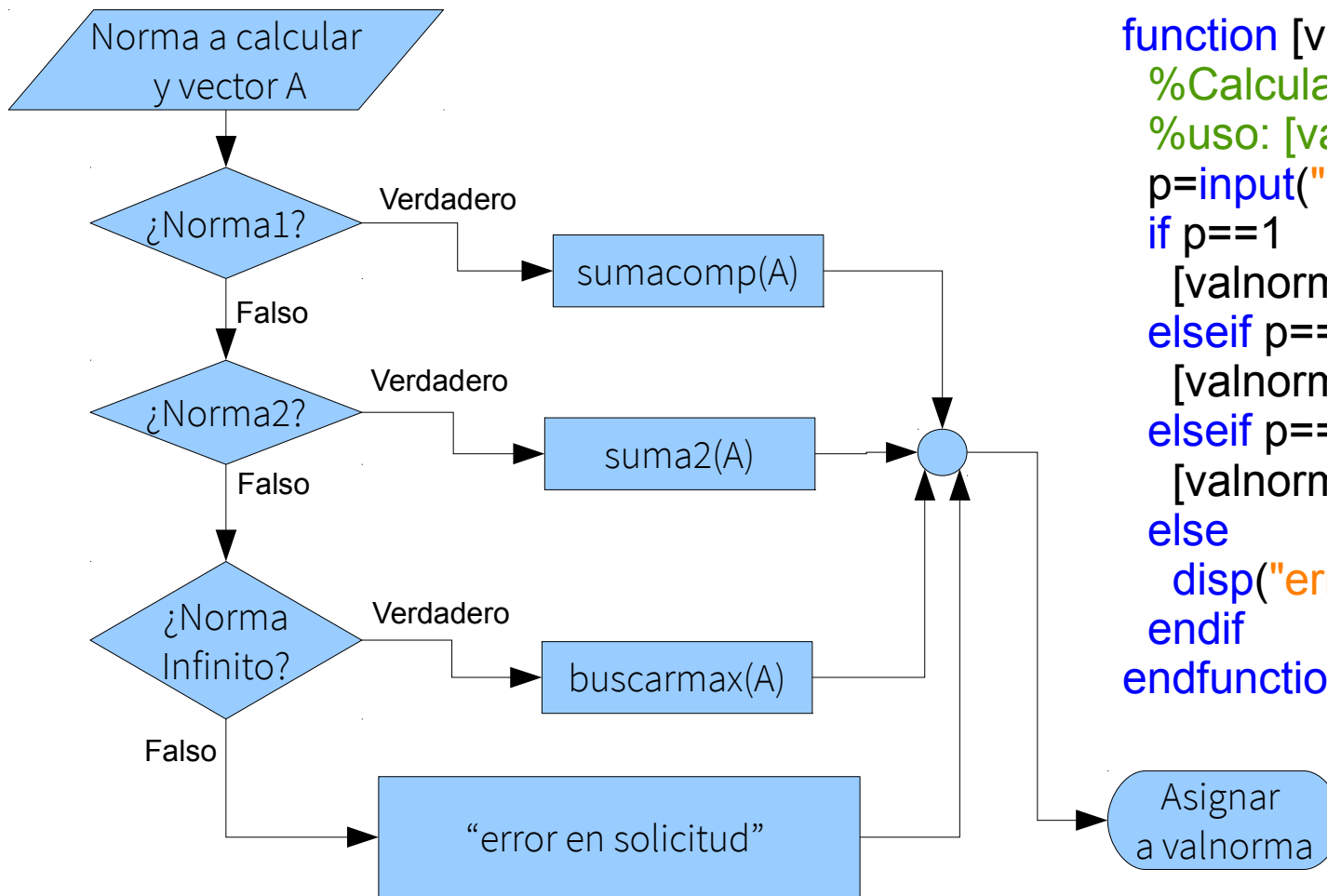
## Cálculo de suma de sus componentes

```
function [valsuma]=sumacomp(A)
    %suma valores del vector A
    valsuma=A(1);
    N=length(A); %ver tamaño del vector
    for i=2:N
        valsuma=valsuma+A(i);
    endfor
endfunction
```

```
function [valor]=suma2(A)
    %?????????
    valor=A(1)^2;
    N=length(A);
    for i=2:N
        valor=valor+A(i)^2;
    endfor
    valor=sqrt(valor);
endfunction
```

# Operaciones sobre vectores

## Normas



```
function [valnorma]=calcnorma(A)
%Calcula norma 1,2 o Inf
%uso: [valnorma]=calcnorma(A)
p=input("Ingrese norma a calcular:");
if p==1
[valnorma]=sumacomp(A);
elseif p==2
[valnorma]=suma2(A);
elseif p==Inf
[valnorma]=buscarmax(A);
else
disp("error en solicitud");
endif
endfunction
```