

CATEDRA INFORMATICA CBI

FACET – UNT

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION

CARRERAS:

ING INFORMATICA

ING AZUCARERA

ING QUIMICA

ING INDUSTRIAL

ING MECANICA

AGRIMENSURA

ING GEODESICA Y GEOFISICA

2023



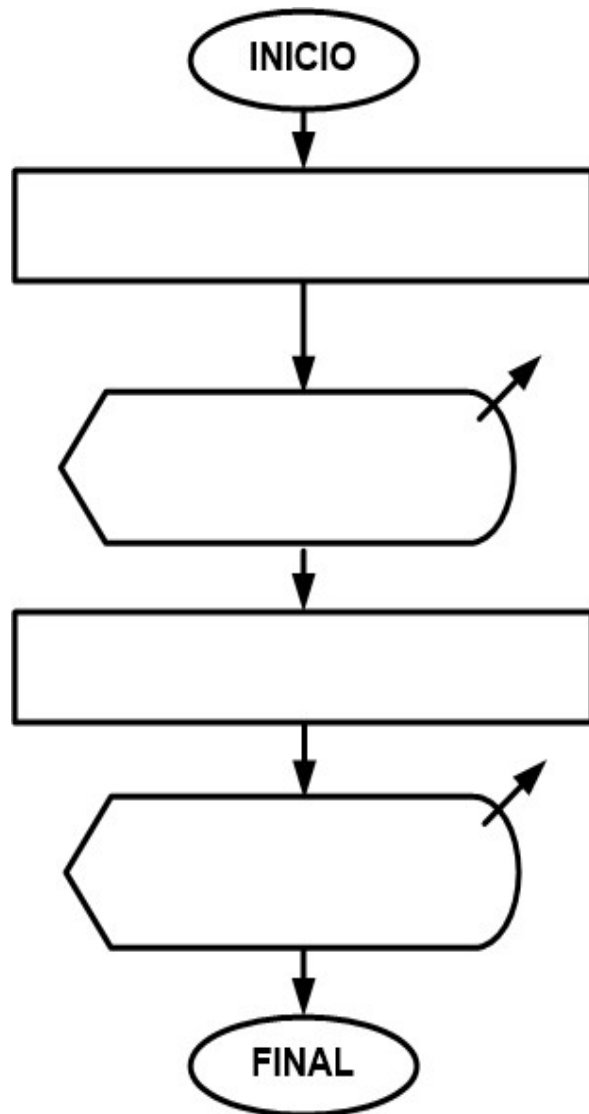
**UNIVERSIDAD
NACIONAL
DE TUCUMÁN**



Funciones I en lenguaje C

CATEDRA INFORMATICA CBI FACET - UNT

Estructura de un programa en lenguaje C, sin funciones



/*Comienzo del programa principal o Función principal*/
#include...

int main() **{/*main() es una función*/**

...

Bloque de código 1

...

Bloque de código 2

...

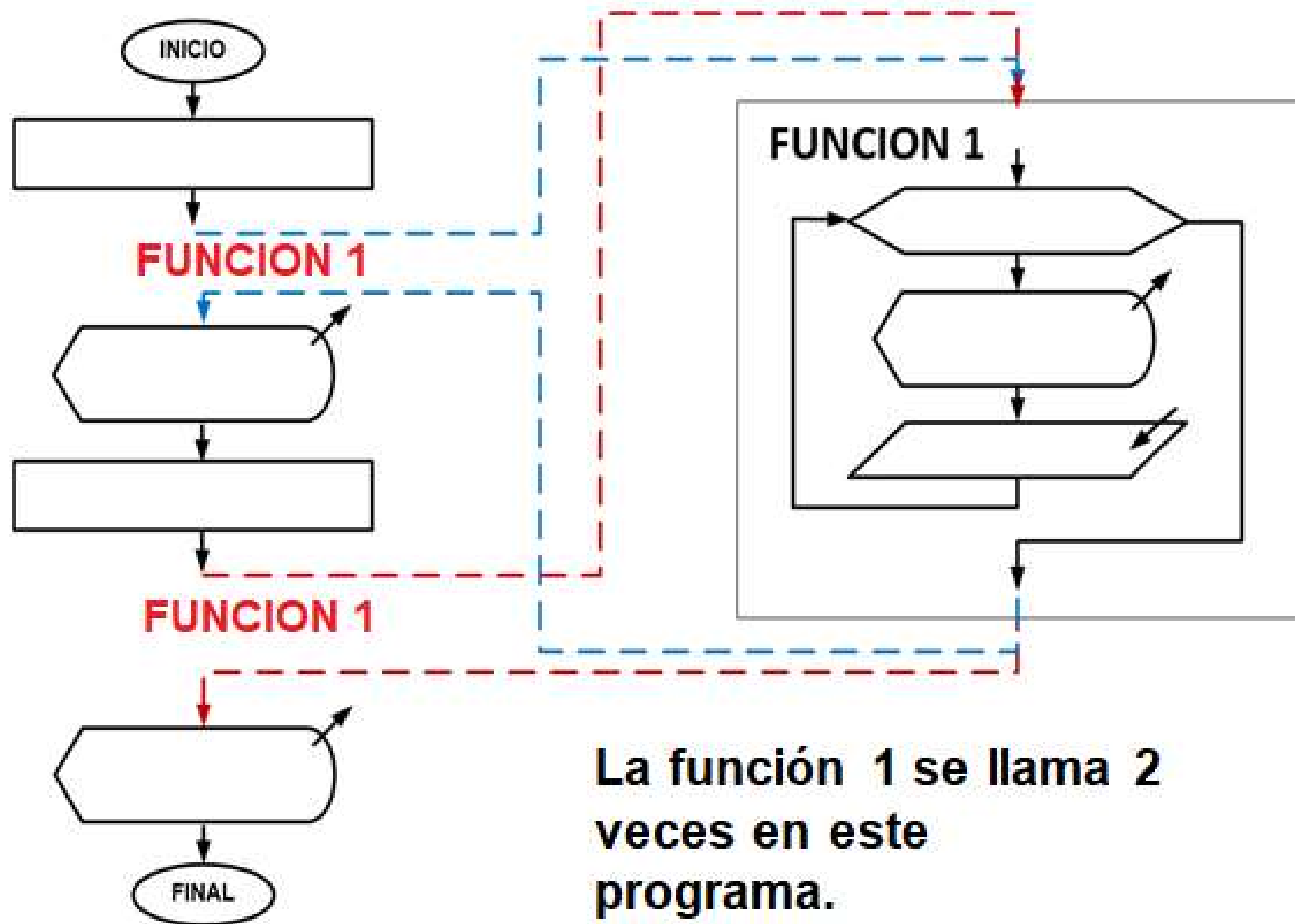
return 0;

}

/*Final de la Función principal*/

CATEDRA INFORMATICA CBI FACET - UNT

Estructura de un Programa , utilizando funciones



CATEDRA INFORMATICA CBI FACET - UNT

Estructura de un Programa en C, usando funciones

*/*Programa principal*/*

```
#include...  
/*Sección de declaraciones de  
funciones*/  
tipo_dato funcion1  
(lista_parametros_1);  
tipo_dato funcion2  
(lista_parametros_2);  
int main() { /* función principal*/  
...  
/*Llamada a la funcion1*/  
funcion1 (lista_parametros_1);  
...  
/*Llamada a la funcion2*/  
funcion2 (lista_parametros_2);  
...  
return 0;  
} /*Fin función principal*/
```

*/*Definición de la función 1*/*

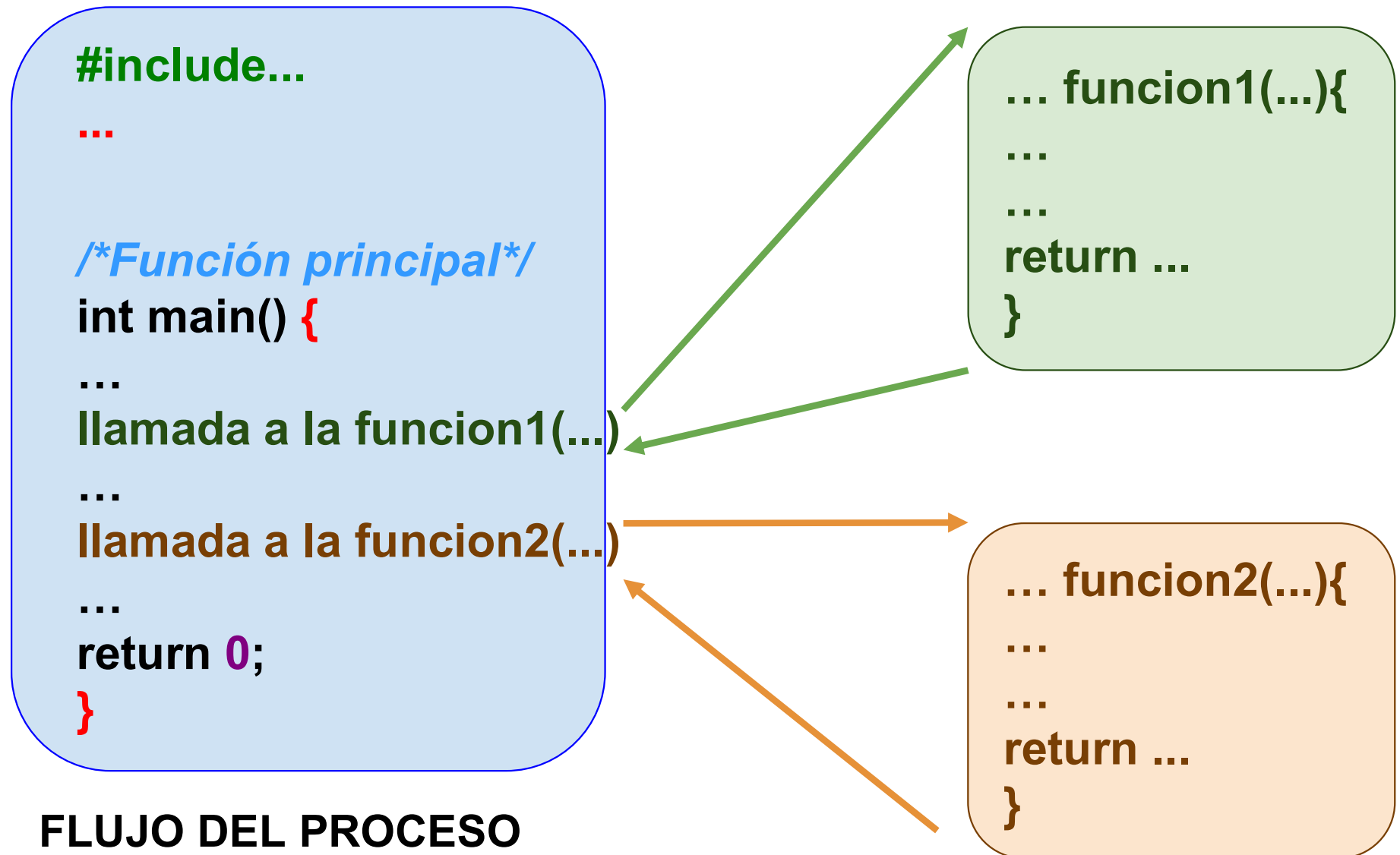
```
tipo_dato  
funcion1(lista_parametros  
_1) {  
Cuerpo de la función 1  
} /* Fin bloque de  
funcion1*/
```

*/*Definición de la función 2*/*

```
tipo_dato  
funcion2(lista_parametros  
_2) {  
Cuerpo de la función 2  
} /* Fin bloque de  
funcion2*/
```

CATEDRA INFORMATICA CBI FACET - UNT

Estructura de un Programa en C, usando funciones



¿Qué es una función?

- **main() es el programa o la función principal**
- **Una función en C, es un fragmento de código que se puede llamar desde cualquier punto de un programa o desde otra función.**
- **Realizan una tarea específica y se llama o invoca como si fuera un comando.**

¿Qué es una función?

- En C podemos diferenciar entre dos tipos de funciones:
 - a) funciones predefinidas o estándares.
 - b) funciones definidas por el usuario.
- La idea es reutilizar una función, cuando se utiliza en diferentes partes del programa, o también se la puede utilizar en otros programas.

Funciones predefinidas o estándares

Las funciones predefinidas están incluidas en bibliotecas. La biblioteca estándar consta de varios archivos cabecera

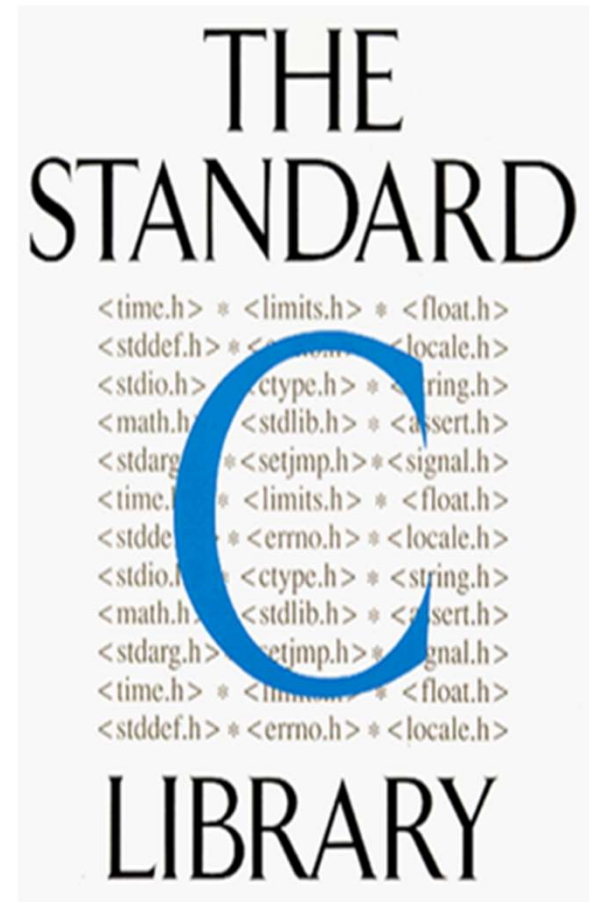
- Entre las más utilizadas, tenemos:

stdio.h ← printf, scanf, etc.

ctype.h ← toupper, isdigit, etc.

string.h ← strcmp, strcpy, strcat, etc.

stdlib.h ← system, exit, atof, atoi, etc.



Funciones definidas por el usuario

- Subprograma, que cuando se lo invoca, realiza una tarea específica, determinada por el programador.
- Tratar de estandarizar el código empleado.
- Basado en la teoría de «**Divide y vencerás**»
- Pueden ser probadas o modificadas.
- Basado también en el concepto de diseño descendente (TOP-DOWN)



Formato de funciones definidas por el usuario

- **DECLARACIÓN** (debajo de los header)
- **DEFINICIÓN** (al final del Main)
- **LLAMADA** (dentro del Main, muchas veces)

Declaración de una función en C (cabecera)

La Declaración de la función consta de las siguientes partes:

- Tipo de dato que retorna la función
- Nombre de la función
- Lista de parámetros que recibe, encerrados entre paréntesis
- Siempre debe finalizar con un punto y coma

Ejemplos:

- `void presentacion (void);`
- `int es_numero_par (int b);`
- `int es_numero_primo (int x);`
- `void mostrar (int x);`
- `float suma (int x, float y);`

Formato de funciones definidas por el usuario

- **DECLARACIÓN** (debajo de los header)
- **DEFINICIÓN** (al final del Main)
- **LLAMADA** (dentro del Main, muchas veces)

DEFINICIÓN DE UNA FUNCIÓN EN C

Debe ser indicativo de la tarea que realiza

Declaración implícita de variables que toman valores del programa que lo invoca

tipo_de_dato_devuelto **nombre_de_la_función** (**lista de parámetros**) {

cuerpo de la función

return **expresión**;

}

Puede retornar cualquier tipo de dato estándar en C: void, int, char, float, etc. O definido por el usuario, a excepción de los arreglos

Formato de funciones definidas por el usuario

- **DECLARACIÓN** (debajo de los header)
- **DEFINICIÓN** (al final del Main)
- **LLAMADA** (dentro del Main, muchas veces)

LLAMADA DE UNA FUNCIÓN EN C

Es la ubicación dentro del programa, en donde se invoca a la función

Debe haber correspondencia con:

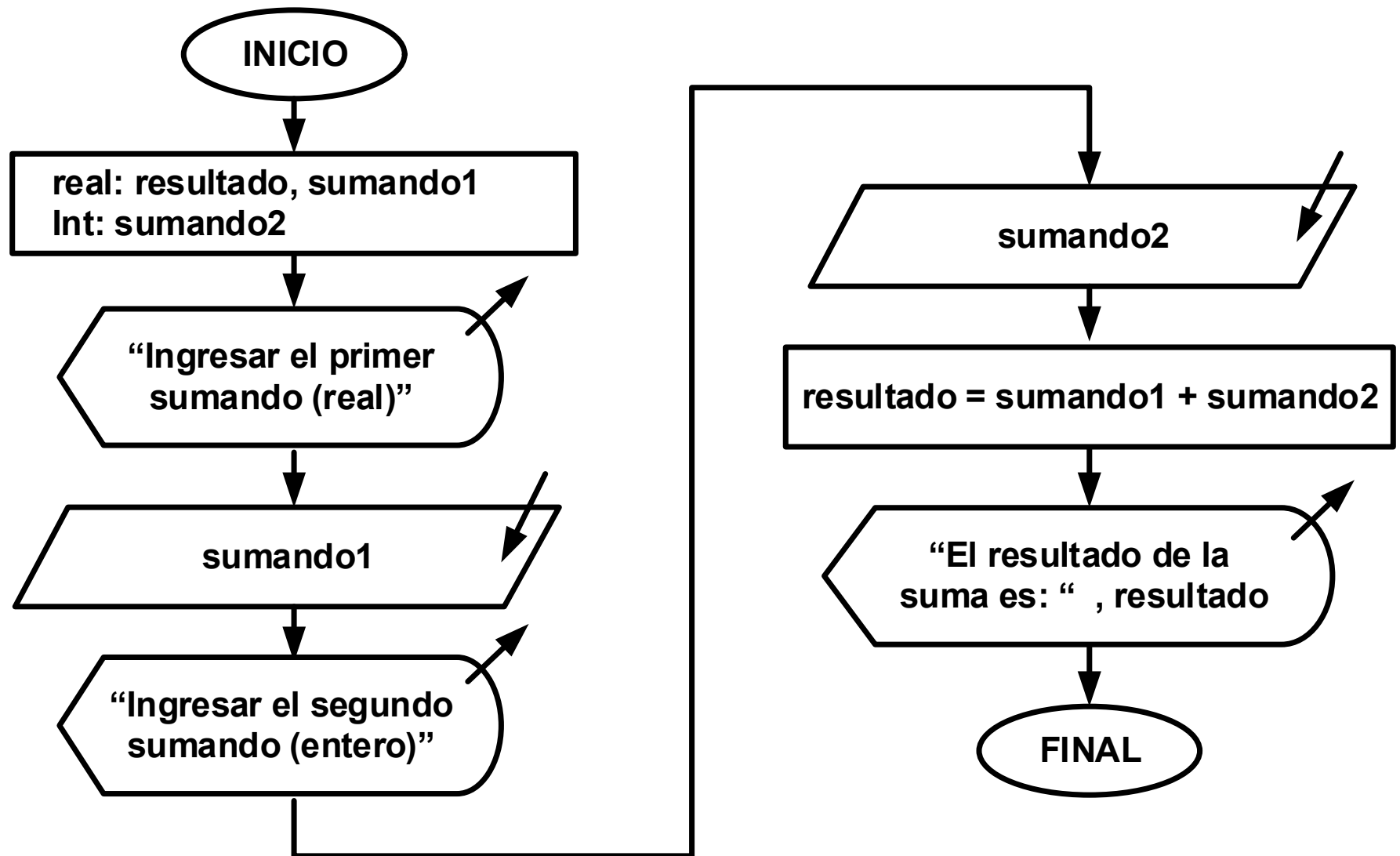
- El tipo de dato que devuelve o retorna la función.
- El nombre de la función con el nombre definido anteriormente.
- La cantidad y el tipo de los argumentos o parámetros que se pasan a la función.

Ejemplos

Declaración	Llamada	
	Bien invocada	Mal invocada
<code>void presentacion(void);</code>	<code>presentacion();</code>	<code>presentacion(x);</code>
<code>int numero_par(int a);</code>	<code>x=numero_par(a);</code>	<code>numero_par(int a);</code>
<code>int numero_primo(int x);</code>	<code>p=numero_primo(x);</code>	<code>p=Numero_primo(x);</code>
<code>void mostrar(int x);</code>	<code>mostrar(x);</code>	<code>void mostrar(x);</code>

EJEMPLO 1: SUMA SIN FUNCION

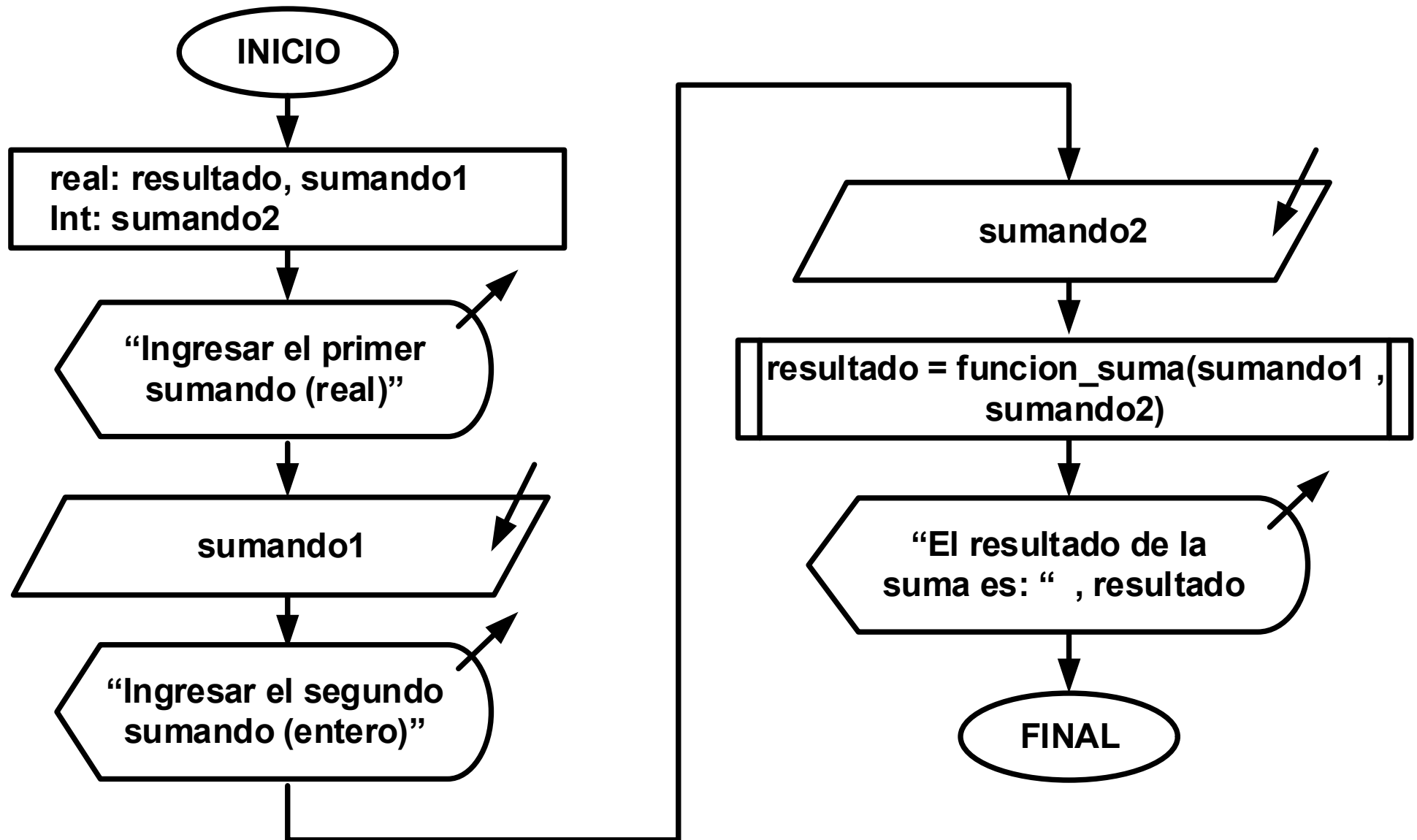
Diagrama de Flujo



EJEMPLO 1: SUMA SIN FUNCION CODIGO

```
1  /*Ejemplo 1: Realizar un programa que permita sumar un
2  número real con otro número entero, luego presentar por
3  pantalla el resultado de la suma.*/
4  #include <stdio.h>
5  int main() { /*función principal o programa principal*/
6      float resultado, sumando1; /*declaración de variables*/
7      int sumando2; /*Presentación de mensajes e ingreso de datos*/
8      printf("Ingresar el primer sumando (real): ");
9      scanf("%f",&sumando1);
10     printf("Ingresar el segundo sumando (entero): ");
11     scanf("%d",&sumando2);
12     resultado=sumando1+sumando2; /*Procesamiento de los datos*/
13     /*Presentación del resultado por pantalla*/
14     printf("El resultado de la suma es: %f \n\n", resultado);
15     return 0;
16 }
```

EJEMPLO 2: SUMA CON FUNCION - Diagrama de Flujo main()



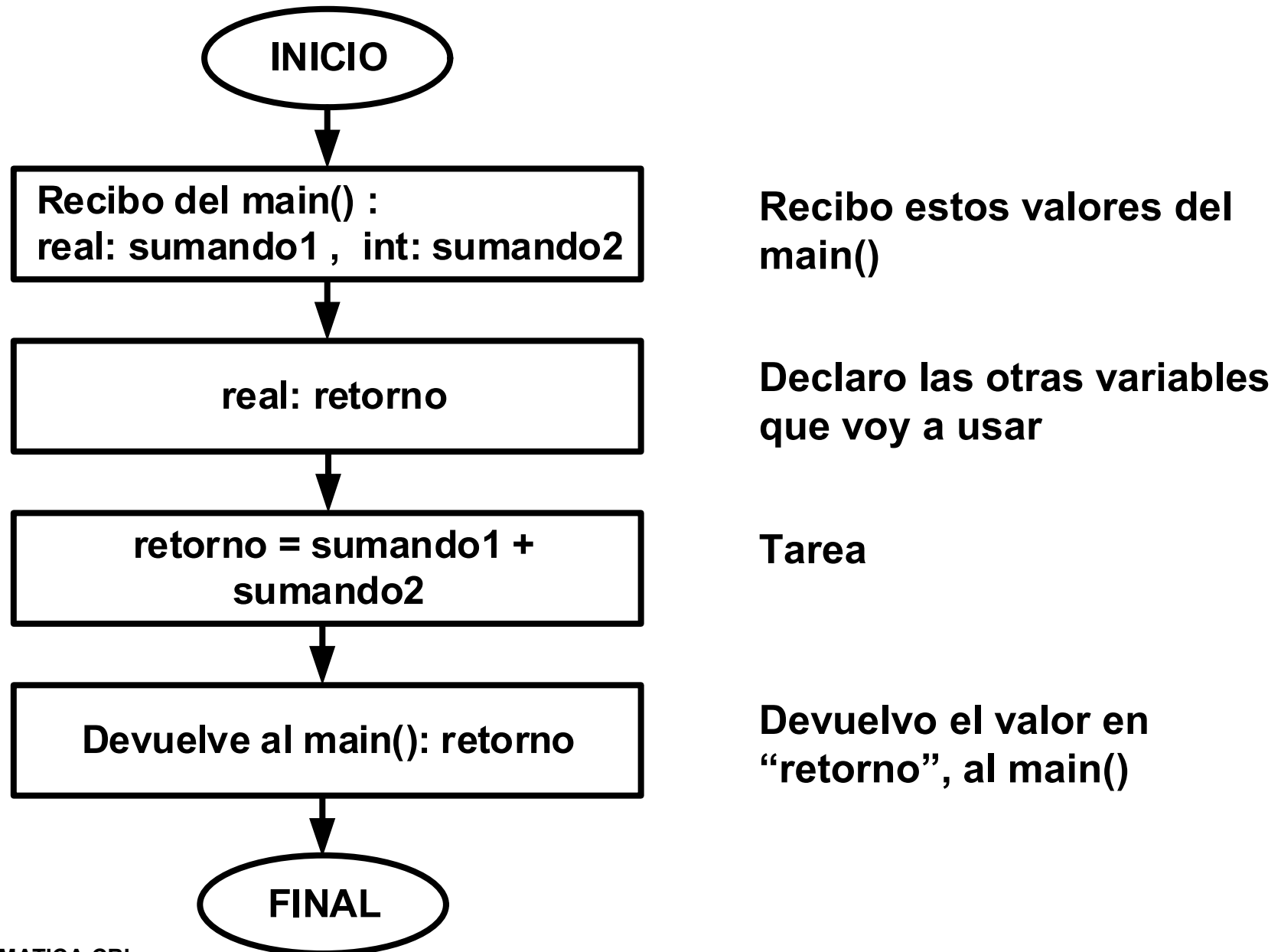
EJEMPLO 2: SUMA CON FUNCION - CODIGO

main()

```
1  /*Ejemplo 2: Realizar un programa que permita sumar un número real
2  con otro número entero, luego presentar por pantalla el resultado
3  de la suma. Utilizando una función */
4  #include <stdio.h>
5  float funcion_suma (float sumando1, int sumando2); //Declara función
6  int main() { //Comienzo de la función principal
7      float resultado, sumando1; //declaración de variables
8      int sumando2; //Presentación de mensajes e ingreso de datos
9      printf("Ingresar el primer sumando (real): ");
10     scanf("%f",&sumando1);
11     printf("Ingresar el segundo sumando (entero): ");
12     scanf("%d",&sumando2); //en 'resultado' se copiará el valor retornado
13     resultado=funcion_suma(sumando1, sumando2); //Llamada a la función
14     //Presentación del resultado por pantalla
15     printf("El resultado de la suma es: %f \n\n", resultado);
16     return 0;
17 }
```

continua

EJEMPLO 2: SUMA CON FUNCION - Diagrama de Flujo funcion_suma()



EJEMPLO 2: SUMA CON FUNCION - CODIGO

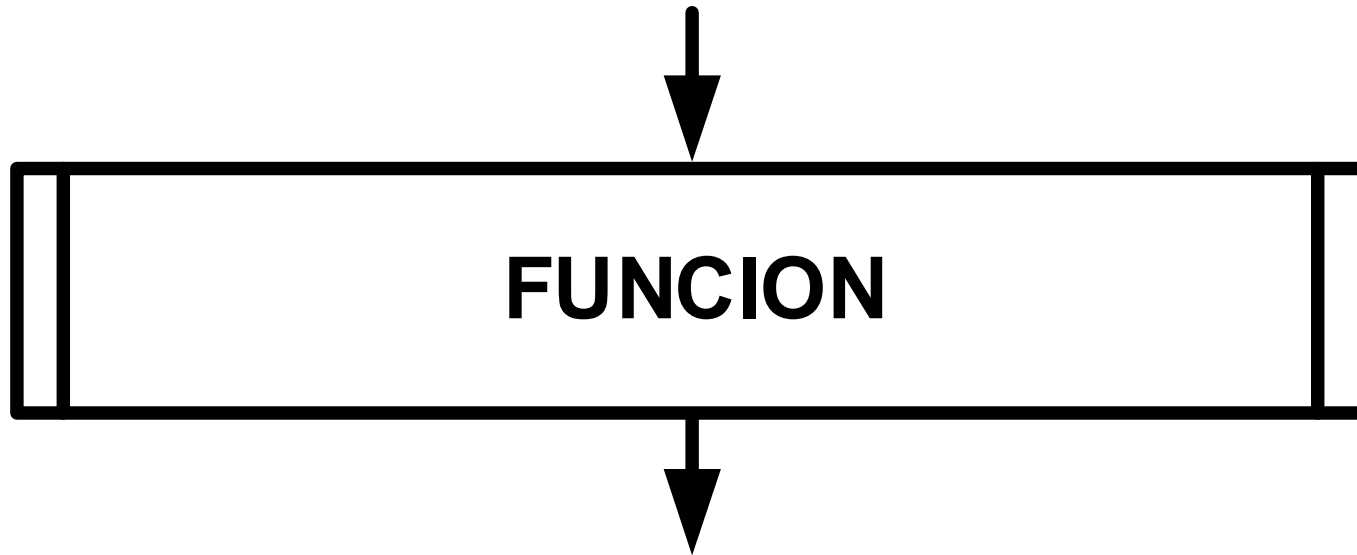
funcion_suma()

continuación

```
18 //Comienzo de la función suma
19 float funcion_suma (float sumando1, int sumando2){//Definición de función
20     float retorno; //Declaración de variables locales a la función suma
21     //Procesamiento de los datos
22     retorno=sumando1+sumando2; //el resultado se asigna a retorno
23     return (retorno);
24 } //cuando finaliza, devuelve el valor contenido en retorno
25 //Fin de la función suma
```

final

SIMBOLO DE FUNCION

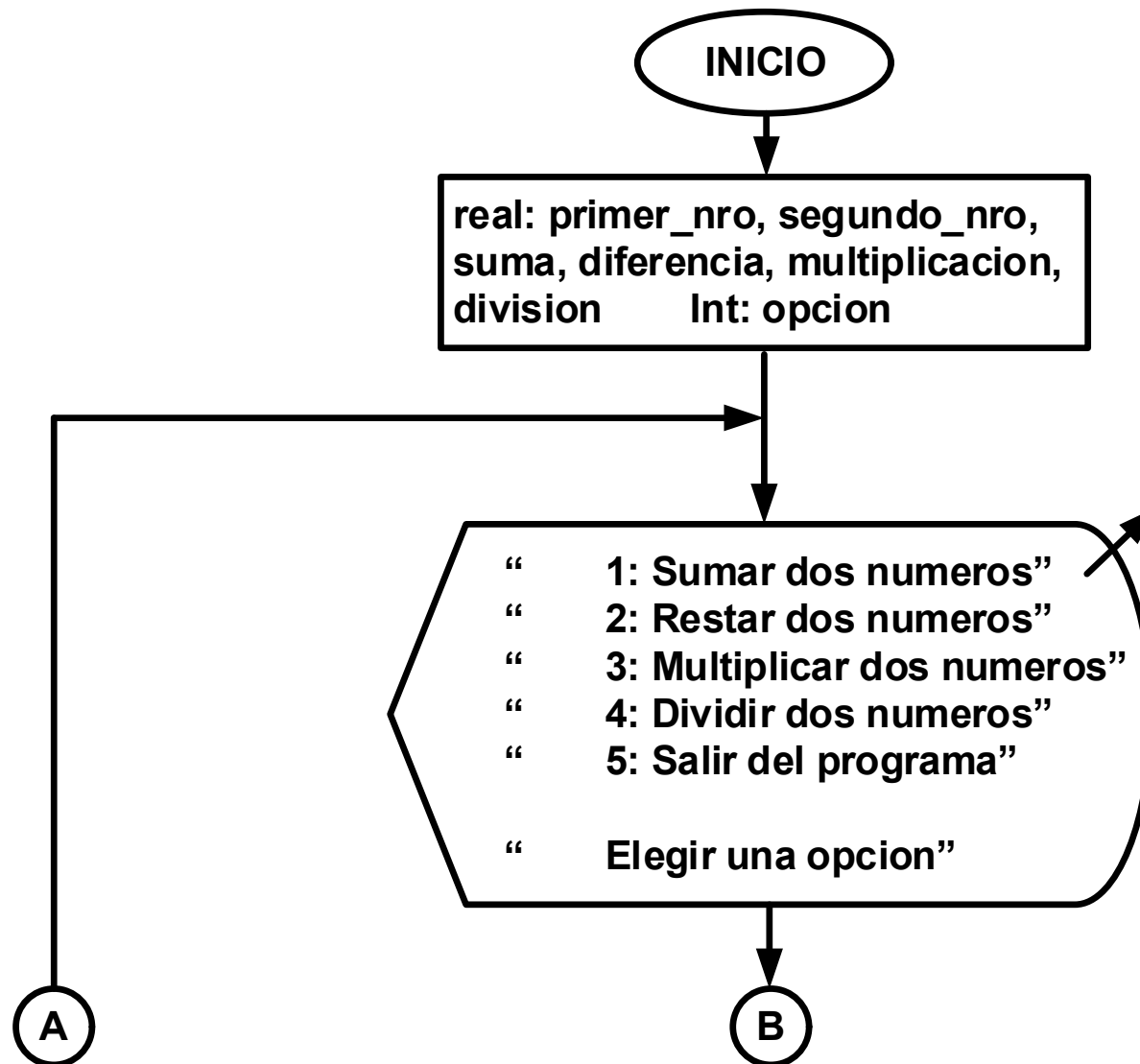


- **UNA ENTRADA Y UNA SALIDA**
- **PUEDE SER TODO UN PROGRAMA COMPLEJO**
- **SOLUCIONA UN PROBLEMA ESPECIFICO**
- **SE PUEDE REUSAR LAS VECES QUE SEA NECESARIO**

CATEDRA INFORMATICA CBI FACET - UNT

EJEMPLO 3: CON MENU Y FUNCIONES

parte 1: Enunciado, Declaración de librerías, Declaración de funciones, main(), declaración de variables, abre el DO-WHILE, muestra menú



CATEDRA INFORMATICA CBI FACET - UNT

EJEMPLO 3: CON MENU Y FUNCIONES

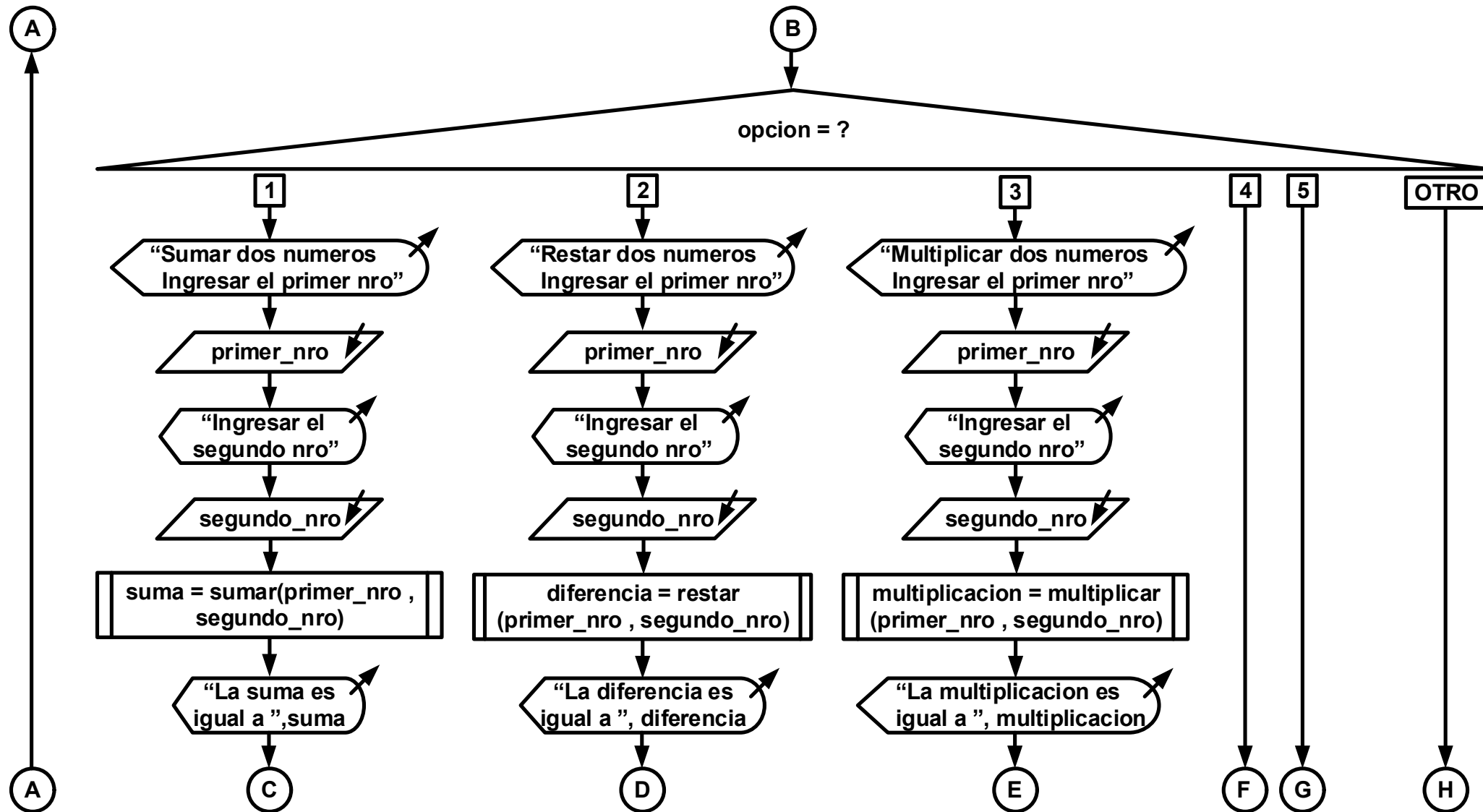
parte 1: *Enunciado, Declaración de librerías, Declaración de funciones, main(), declaración de variables*

```
1  /* Ejemplo 3: Enunciado: Realizar un programa que muestre un menú que
2  contenga las siguientes opciones:
3      1: Sumar dos números
4      2: Restar dos números
5      3: Multiplicar dos números
6      4: Dividir dos números
7      5: Salir del programa
8  Luego deberá realizar la operación seleccionada utilizando funciones*/
9  #include<stdio.h> //Declaración de librerías
10 float sumar(float a, float b); //Declaración de las funciones
11 float restar(float a, float b);
12 float multiplicar(float a, float b);
13 float dividir(float a);
14 ☐ int main(){ //Función principal main()
15     int opcion; //Declaración de variables
16     float primer_nro,segundo_nro,suma,diferencia,multiplicacion,division;
```

continua

EJEMPLO 3: CON MENU Y FUNCIONES

parte 2: SWITCH, las 3 primeras opciones del menú



CATEDRA INFORMATICA CBI FACET - UNT

parte 2: Menú, las 2 primeras opciones del SWITCH

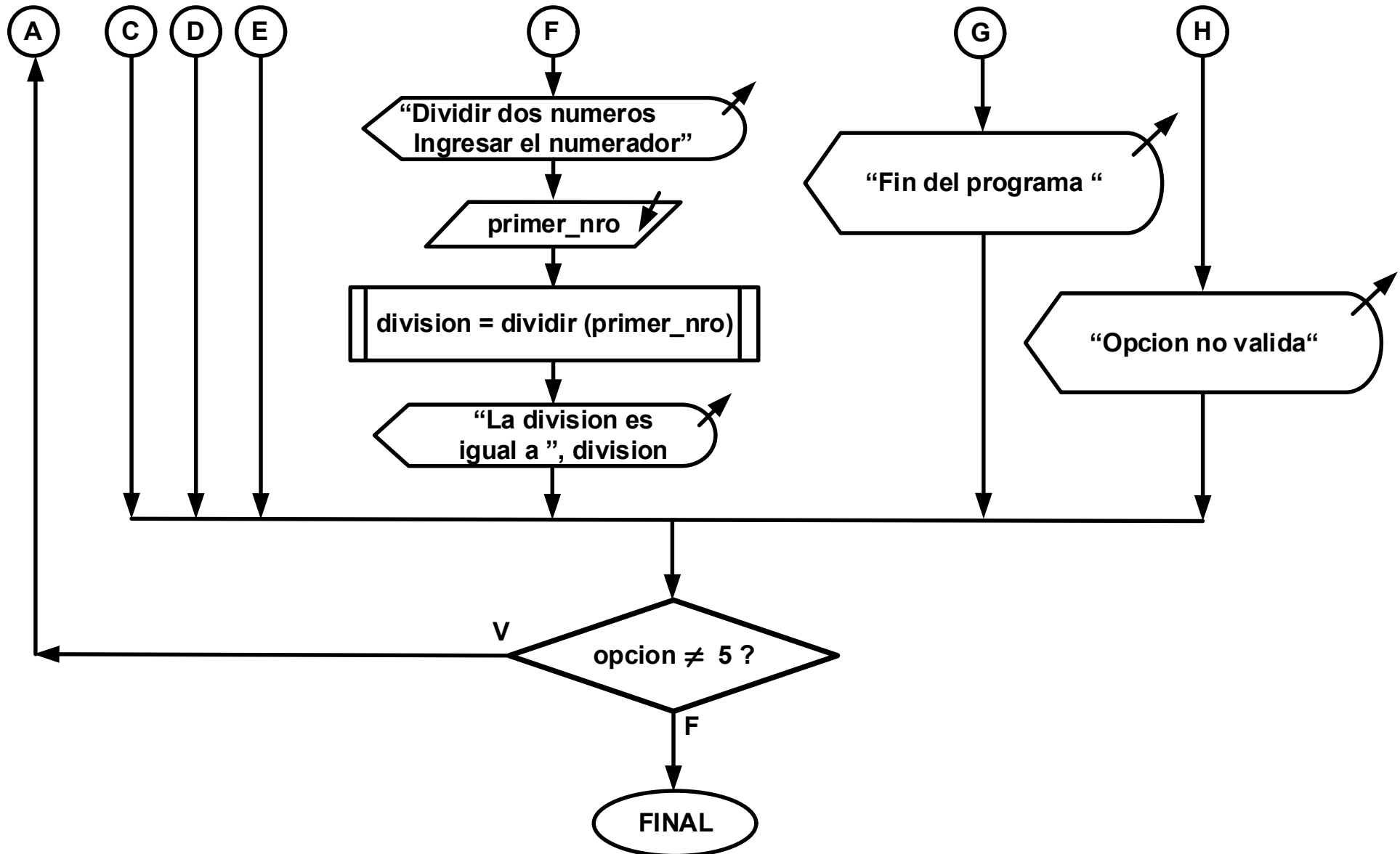
continuación

```
17 do{ //Muestra el menú por pantalla
18     printf("\n\n\t\t1: Sumar dos numeros\n"); //\t\t agrega
19     printf("\t\t2: Restar dos numeros\n"); // margen izquierdo
20     printf("\t\t3: Multiplicar dos numeros\n");
21     printf("\t\t4: Dividir dos numeros\n");
22     printf("\t\t5: Salir del programa\n\n");
23     printf("\t\tElegir una opcion: ");
24     scanf("%d",&opcion);
25     switch(opcion){ /*Comienzo de la Estructura switch*/
26         case 1: printf("\t\t Sumar dos numeros\n");
27                 printf("Ingresar el primer nro: ");
28                 scanf("%f",&primer_nro);
29                 printf("Ingresar el segundo nro: ");
30                 scanf("%f",&segundo_nro);
31                 suma=sumar(primer_nro,segundo_nro); //invoca la funcion 'sumar'
32                 printf("\t La suma es igual a %.2f \n",suma);
33                 break;
34         case 2: printf("\t\t Restar dos numeros\n");
35                 printf("Ingresar el primer nro: ");
36                 scanf("%f",&primer_nro);
37                 printf("Ingresar el segundo nro: ");
38                 scanf("%f",&segundo_nro);
39                 diferencia=restar(primer_nro,segundo_nro); //invoca la funcion 'restar'
40                 printf("\tLa diferencia es igual a %.2f \n",diferencia);
41                 break;
```

continua

EJEMPLO 3: CON MENU Y FUNCIONES

parte 3: Las últimas 3 opciones del SWITCH, cierra el SWITCH, cierra el DO-WHILE, cierra el main()



CATEDRA INFORMATICA CBI FACET - UNT

parte 3: Las ultimas 3 opciones, y la opción OTROS, cierra el SWITCH, cierra el main()

continuación

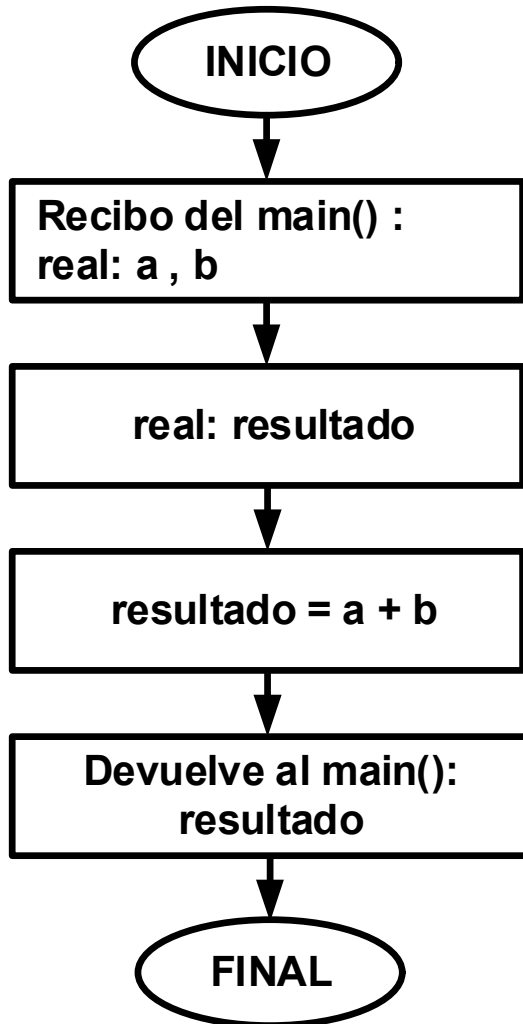
```
42     case 3: printf("\t\t Multiplicar dos numeros\n");
43             printf("Ingresar el primer nro: ");
44             scanf("%f",&primer_nro);
45             printf("Ingresar el segundo nro: ");
46             scanf("%f",&segundo_nro);
47             multiplicacion=multiplicar(primer_nro,segundo_nro); //invoca 'multiplicar'
48             printf("\tLa multiplicacion es igual a %.2f\n",multiplicacion);
49             break;
50     case 4: printf("\t\t Dividir dos numeros\n"); //invoca 'dividir'
51             printf("Ingresar el numerador: ");
52             scanf("%f",&primer_nro);
53             division=dividir(primer_nro); //el denominador se lo ingresa en la función
54             printf("\tLa division es igual a %.2f\n",division);
55             break;
56     case 5: printf("Fin del programa ");
57             break;
58     default:printf("Opcion no valida "); //Fin de la Estructura switch
59 }
60 }while(opcion!=5);
61 return 0;
62 } //Cierra con la llave del main - Fin del programa
```

continua

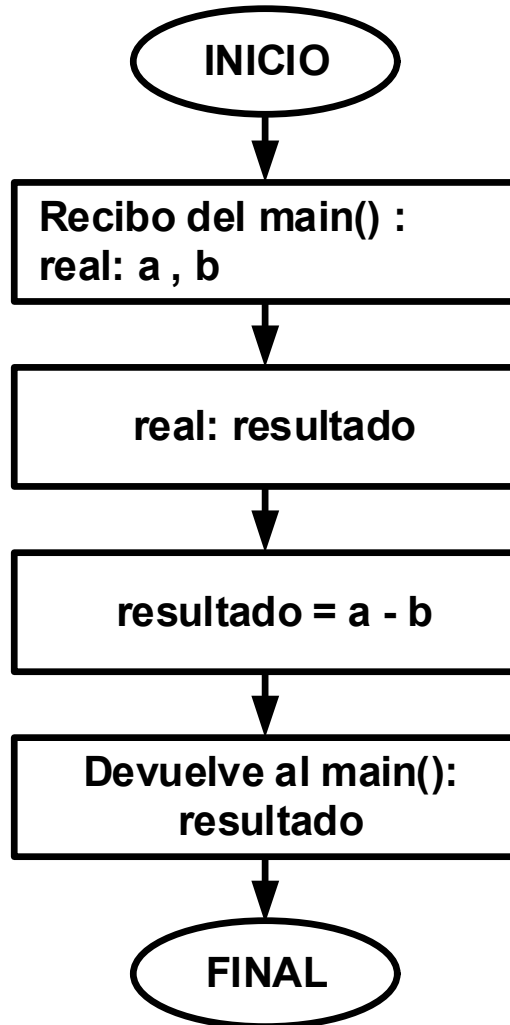
EJEMPLO 3: CON MENU Y FUNCIONES

parte 4: La definición de las 3 primeras funciones

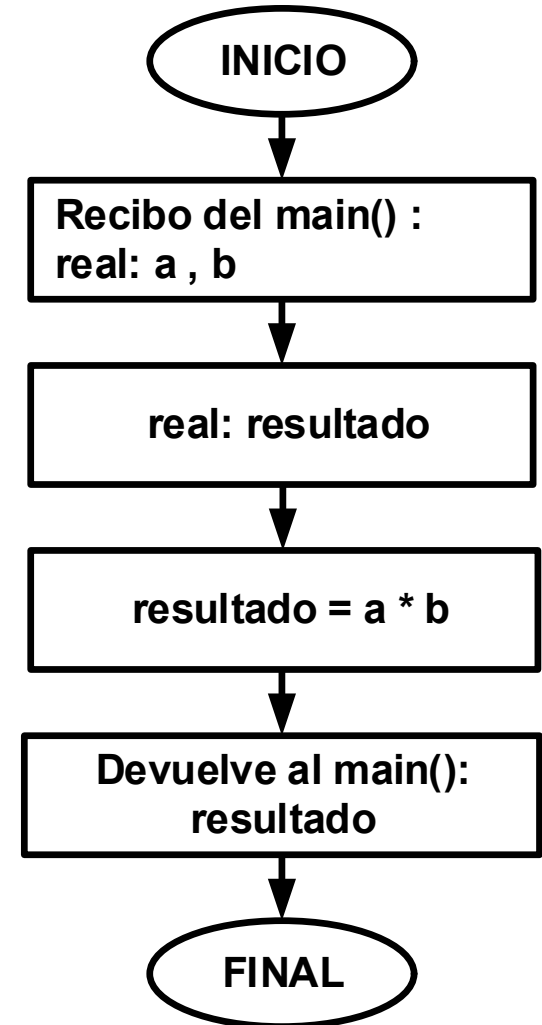
FUNCION SUMAR



FUNCION RESTAR



FUNCION MULTIPLICAR



parte 4: La definición de las 3 primeras funciones

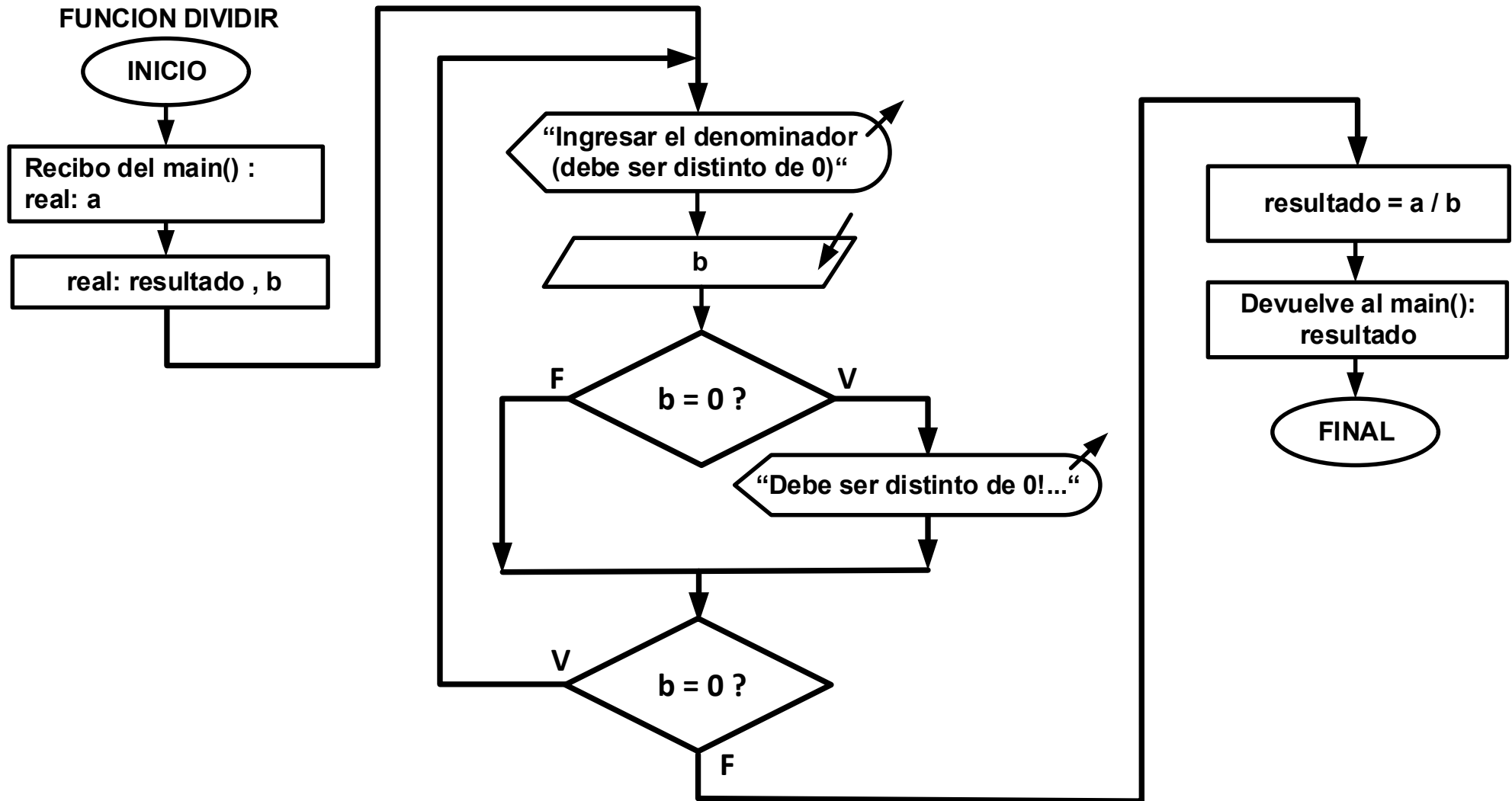
continuación

```
63 //Comienzo de la definición de funciones
64 float sumar(float a, float b) { //Función sumar
65     float resultado;
66     resultado=a+b;
67     return resultado;
68 }
69
70 float restar(float a, float b) { //Función restar
71     float resultado;
72     resultado=a-b;
73     return resultado;
74 }
75
76 float multiplicar(float a, float b) { //Función multiplicar
77     float resultado;
78     resultado=a*b;
79     return resultado;
80 }
```

continua

EJEMPLO 3: CON MENU Y FUNCIONES

parte 5: La definición de la 4° y última función



parte 5: La definición de la 4° y última función

continuación

```
81
82 float dividir(float a) { //Función dividir
83     float resultado, b;
84     do{ //controla que el denominador sea distinto de cero
85         printf("Ingresar el denominador (debe ser distinto de cero:");
86         scanf("%f",&b);
87         if(b==0){
88             printf("\n\nDebe ser distinto de cero!... \n\n");
89         }
90     } while(b==0);
91     resultado=a/b;
92     return resultado;
93 }
```

*/*Fin de la definición de funciones*/*

final