

# FundamentosDeProgramacion

## COMUNIDAD 2

- Molina Flores Roberto Carlos
- Gutiérrez Romero Alberto
- Valdés Osorio Maria Fernanda

## Examen de funciones

### Funcion principal **main()**

El siguiente codigo muestra los programas vistos en clase , implementados como funciones.Mientras en la funcion **main()** se implementa un menu con las opciones a seleccionar que invocan las funciones mostradas a continuacion

#### Librerias utilizadas:

```
#include <stdio.h>
#include <stdlib.h>
#include<math.h>
```

#### Definicion de constantes:

```
#define PI 3.14159265
#define CONSTANTE 20
```

#### Declaracion de funciones:

```
void opciones();
void polar_to_cartesian();
void cartesian_to_polar();
void numero_secreto();
void raices_ecuacion();
void area_perimetro();
void numero_letras();
void suma_de_cuadrados();
void operaciones();
void serie();
void coordenadas_convert();
```

#### Código de la funcion **main()**:

```

int main() {
    do {

        opciones();
        scanf("%d",&opc);

        switch (opc) {
            case 1:
                operaciones();
                break;
            case 2:
                serie();
                break;
            case 3:
                suma_de_cuadrados();
                break;
            case 4:
                coordenadas_convert();
                break;
            case 5:
                numero_secreto();
                break;
            case 6:
                area_perimetro();
                break;
            case 7:
                raices_ecuacion();
                break;
            case 8:
                numero_letras();
                break;
            case 0:
                printf("Hasta pronto");
                break;
            default:
                printf("Valor invalido");
                break;
        }

    } while(opc!=0 && opc>0 && opc<9);

    return 0;
}

```

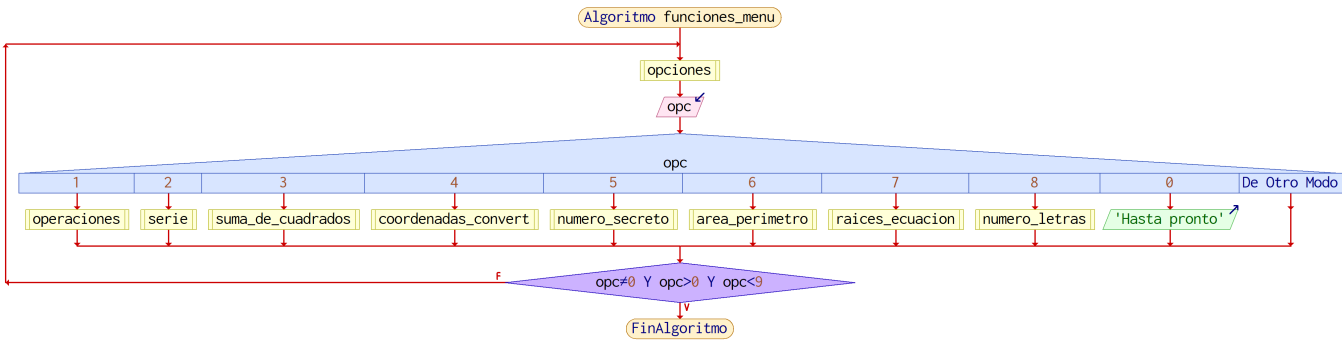
## Pseudocodigo

Algoritmo funciones\_menu  
 Repetir

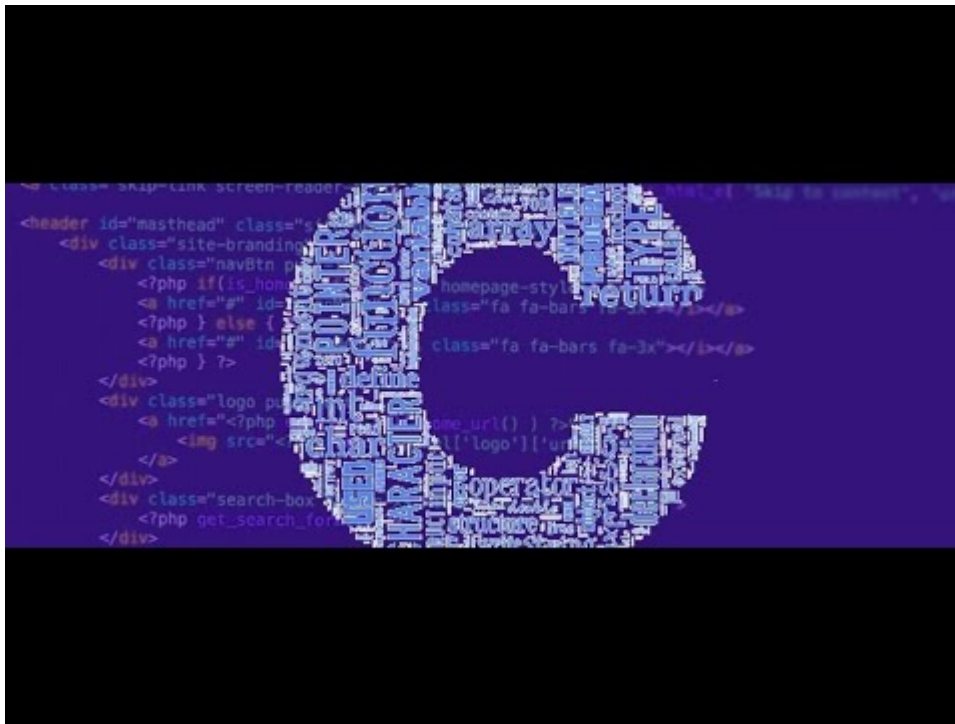
```

opciones
Leer opc
Segun opc  Hacer
    1:
        operaciones
    2:
        serie
    3:
        suma_de_cuadrados
    4:
        coordenadas_convert
    5:
        numero_secreto
    6:
        area_perimetro
    7:
        raices_ecuacion
    8:
        numero_letras
    0:
        Escribir 'Hasta pronto'
FinSegun
Hasta Que opc≠0 Y opc>0 Y opc<9
FinAlgoritmo
    
```

Diagrama de flujo



Ejecución



[Main link](#)

## Funcion **opciones()**

text...

Código de la funcion **opciones()**:

```
void opciones(){

    printf("\nSelecciona una opcion:");
    printf("\n-----");
};
    printf("\n");
    printf("\n\t1-Suma resta multiplicacion, division de numeros enteros");
    printf("\n\t2-Serie 1 1 2 3 5 8 13");
    printf("\n\t3-Desplegar la suma de cuadrados");
    printf("\n\t4-Conversion de coordenadas");
    printf("\n\t5-Numero secreto");
    printf("\n\t6-Calcular area y perímetro");
    printf("\n\t7-Raices de ecuación cuadratica");
    printf("\n\t8-Convertir numero a letras");
    printf("\n\t0-Salir");
    printf("\n\t");
}
```

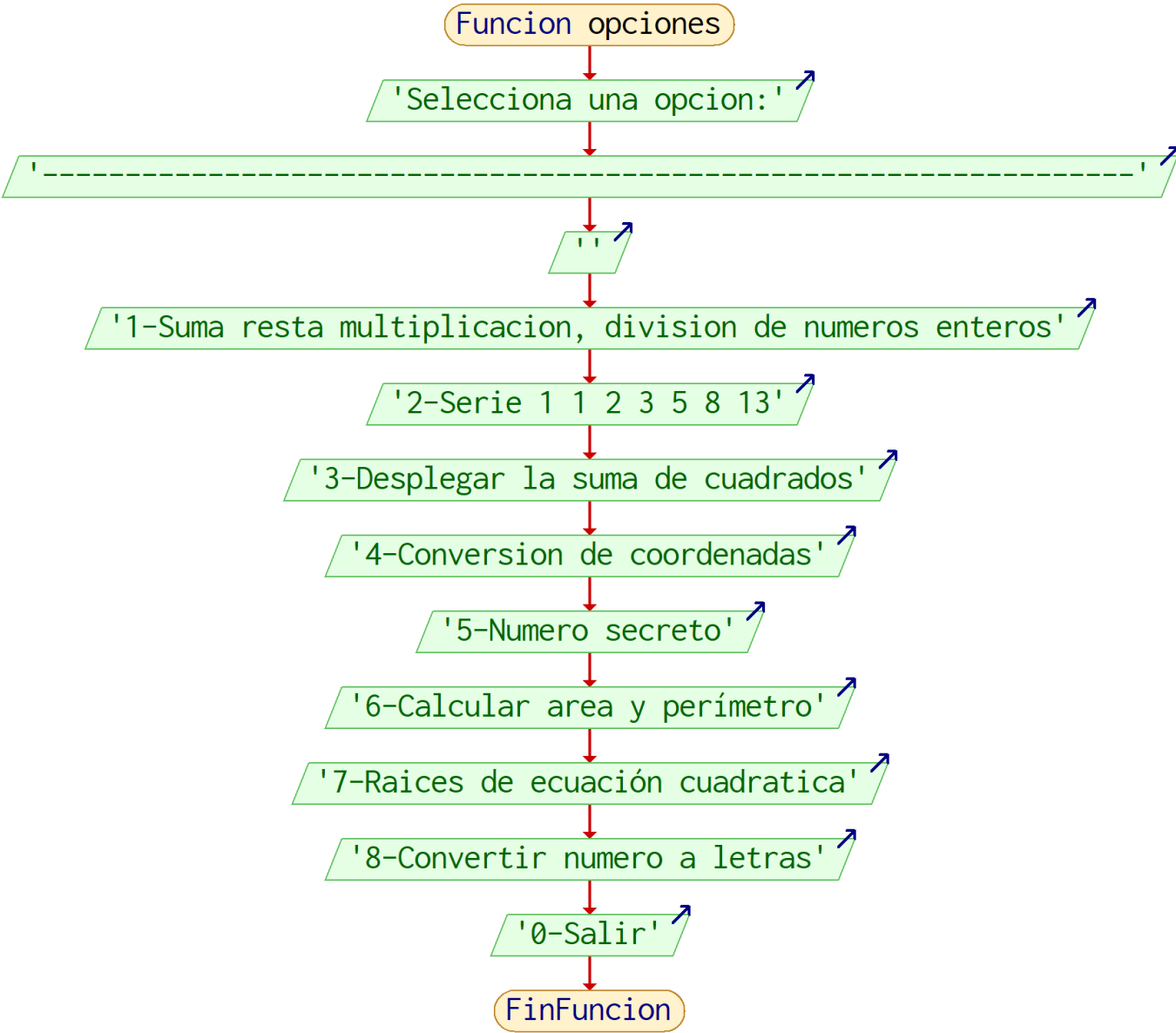
Pseudocodigo

```

Algoritmo funciones_menu
  Repetir
    opciones
    Leer opc
    Segun opc Hacer
      1:
        operaciones
      2:
        serie
      3:
        suma_de_cuadrados
      4:
        coordenadas_convert
      5:
        numero_secreto
      6:
        area_perimetro
      7:
        raices_ecuacion
      8:
        numero_letras
      0:
        Escribir 'Hasta pronto'
    FinSegun
  Hasta Que opc≠0 Y opc>0 Y opc<9
FinAlgoritmo

```

Diagrama de flujo



Ejecución



Opciones link

## Funcion **operaciones()**

text...

Código de la funcion **operaciones()**:

```
void operaciones(){

    float numero1,numero2;
    float suma, multiplicacion, division, resta;
    /* lectura de datos (entradas) */
    printf("\n-----Operaciones-----");
    printf("\nIngrese numero A:\t");
    scanf("%f",&numero1);
    printf("\nIngrese numero B:\t");
    scanf("%f",&numero2);

    suma = numero1 + numero2;
    resta = numero1 - numero2;
    multiplicacion = numero1 * numero2;
    division = numero1 / numero2;
    /* desplegado de datos (salidas) */
    printf("\n-----Resultados-----");
    printf("\nSuma:          \t%.3f \n", suma);
    printf("Resta:          \t%.3f\n", resta);
    printf("Multiplicacion:\t%.3f\n", multiplicacion);
    printf("Division:         \t%.3f\n", division);

}
```

## Pseudocodigo

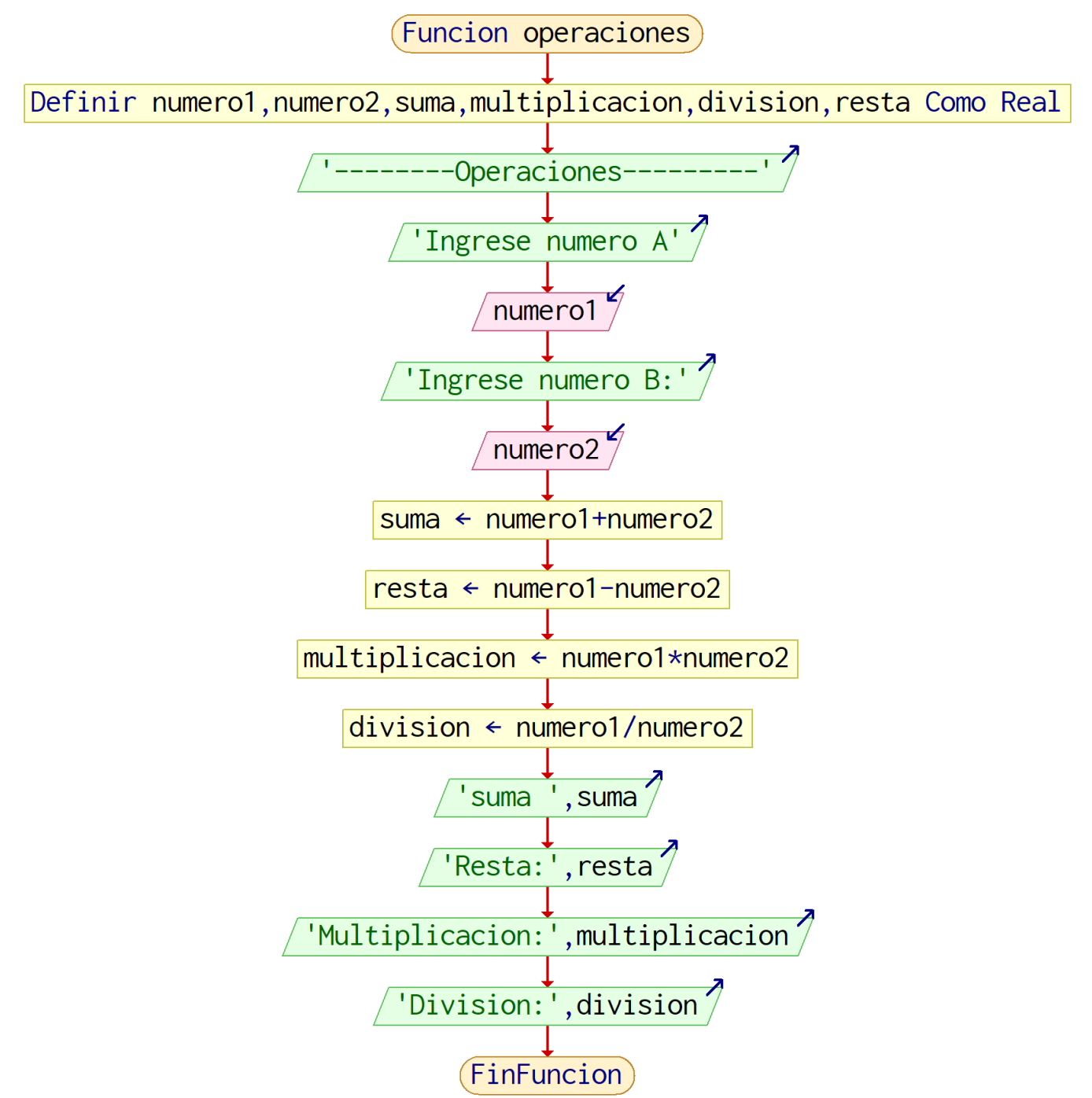
```

Funcion operaciones
    Definir numero1,numero2,suma,multiplicacion,division,resta Como Real
    Escribir '-----Operaciones-----'
    Escribir 'Ingrese numero A'
    Leer numero1
    Escribir 'Ingrese numero B:'
    Leer numero2
    suma ← numero1+numero2
    resta ← numero1-numero2
    multiplicacion ← numero1*numero2
    division ← numero1/numero2
    Escribir 'suma ',suma
    Escribir 'Resta:',resta
    Escribir 'Multiplicacion:',multiplicacion
    Escribir 'Division:',division
FinFuncion

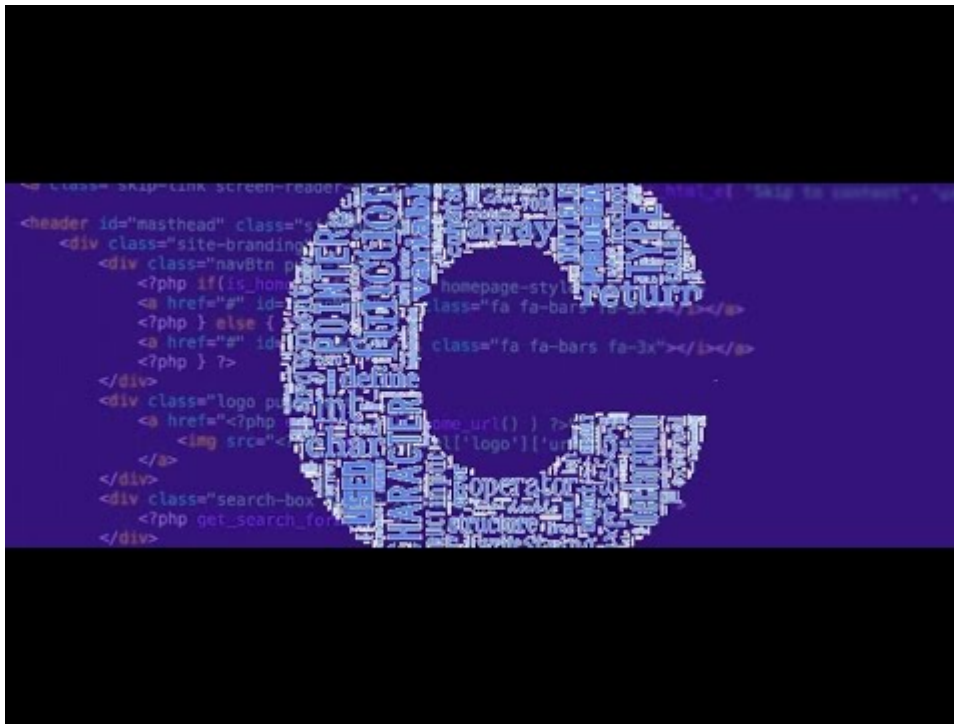
```

## Diagrama de flujo





Ejecución



[Operaciones link](#)

## Funcion **serie()**

text

Código de la funcion **serie()**:

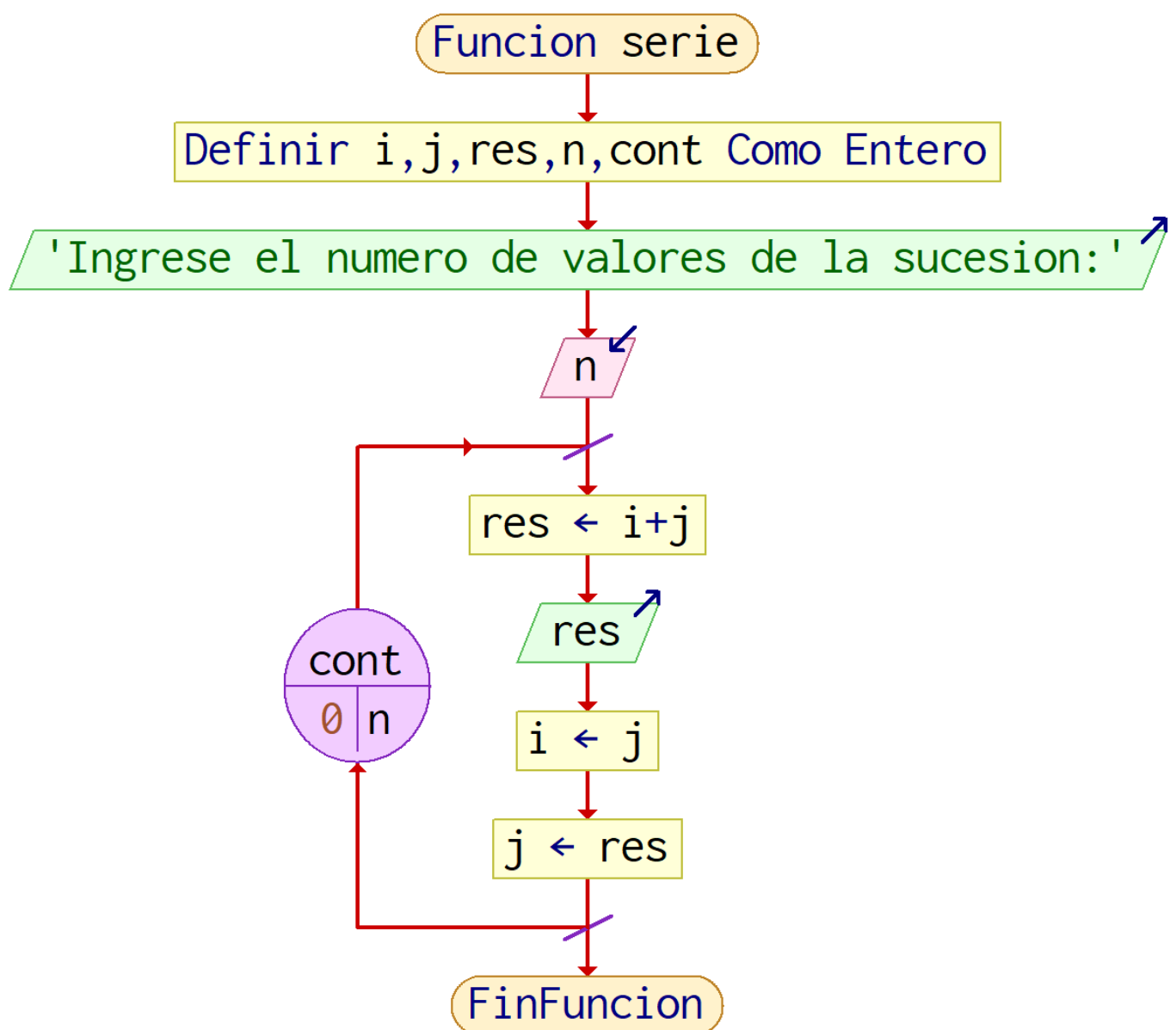
```
void serie(){
    int i, j, serie, N, cont;
    printf("\nIngrese el numero de valores de la sucesion:\t");
    scanf("%d", &N);
    printf("\n");
    i=0;
    j=1;
    for(cont=0; cont<N; cont++)
    {
        serie=i+j;
        printf("%d ", serie);
        i=j;
        j=serie;
    }
    printf("\n");
}
```

Pseudocodigo

```

Funcion serie
  Definir i,j,res,n,cont Como Entero
  Escribir 'Ingrese el numero de valores de la sucesion:'
  Leer n
  Para cont<=0 Hasta n Hacer
    res ← i+j
    Escribir res
    i ← j
    j ← res
  FinPara
FinFuncion
    
```

Diagrama de flujo



Ejecución



Serie link

## Funcion **suma\_de\_cuadrados**

text...

Código de la funcion \*\*\*\*:

```
void suma_de_cuadrados(){
    int x=0;
    int n=0;
    int contador=0;
    int cuadrado=0;
    int impares=0;

    printf("Secuencia de suma de cuadrados\n");
    printf("Ingresa los numeros a mostrar:\t");
    scanf("%d",&n);

    for(contador=1; contador<=n; contador++){
        printf("\nNo. %d = ",contador);
        cuadrado=0;
        x=1;
        do{
            printf("%d",impares);
            if(x<contador) printf("+");
            cuadrado+=impares;
            impares+=2;
            x++;
        }while(x<=contador);
    }
}
```

```

        printf("= %d",cuadrado);
    }
    printf("\n");

}

```

## Pseudocodigo

```

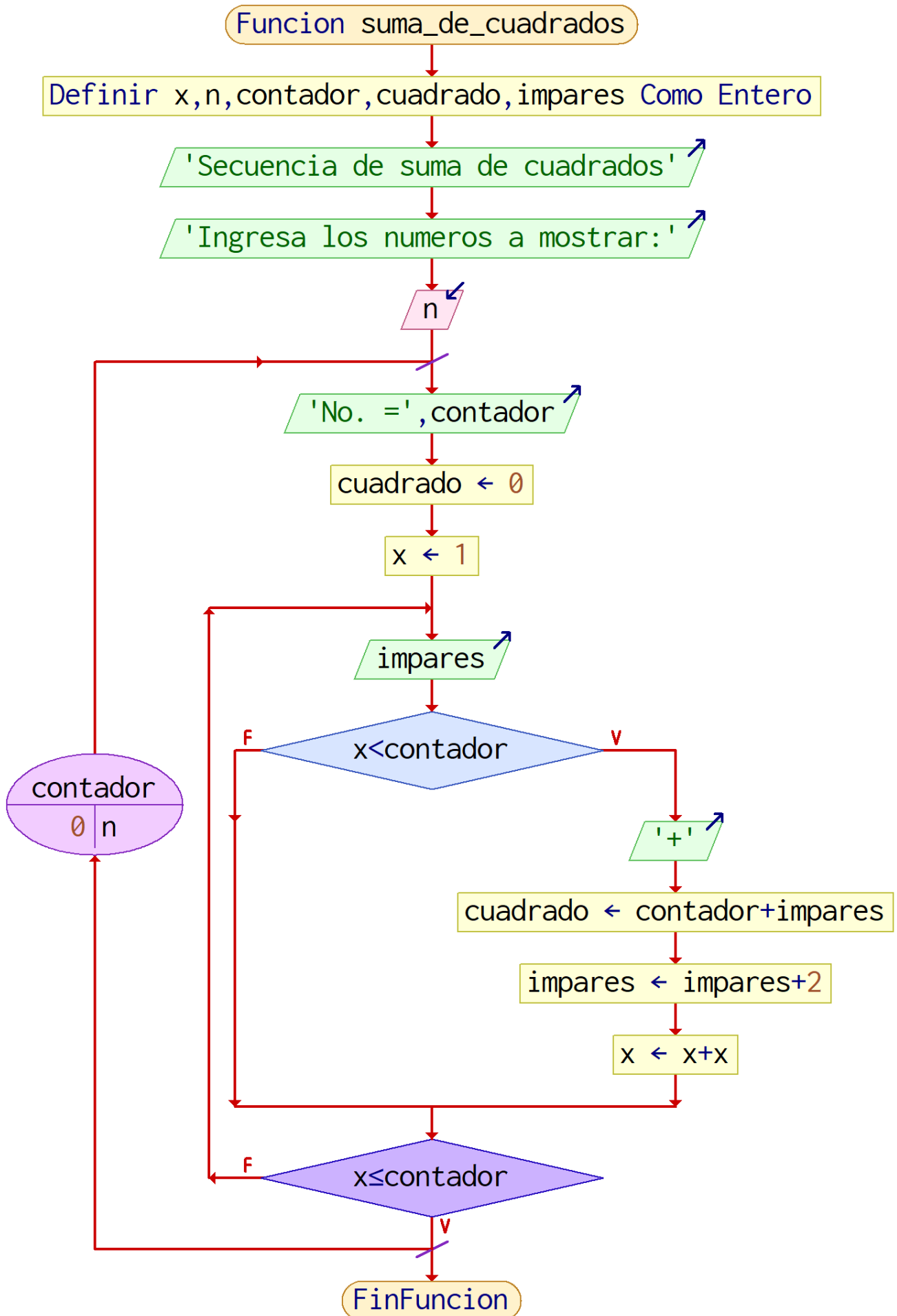
Funcion suma_de_cuadrados
    Definir x,n,contador,cuadrado,impares Como Entero
    Escribir 'Secuencia de suma de cuadrados'
    Escribir 'Ingresa los numeros a mostrar:'
    Leer n

    Para contador<-0 Hasta n Hacer
        Escribir 'No. =',contador
        cuadrado=0
        x=1
        Repetir
            Escribir impares
            si x<contador
                Escribir '+'
                cuadrado=contador+impares
                impares=impares+2
                x=x+x
            FinSi
        Hasta Que x<=contador
    FinPara

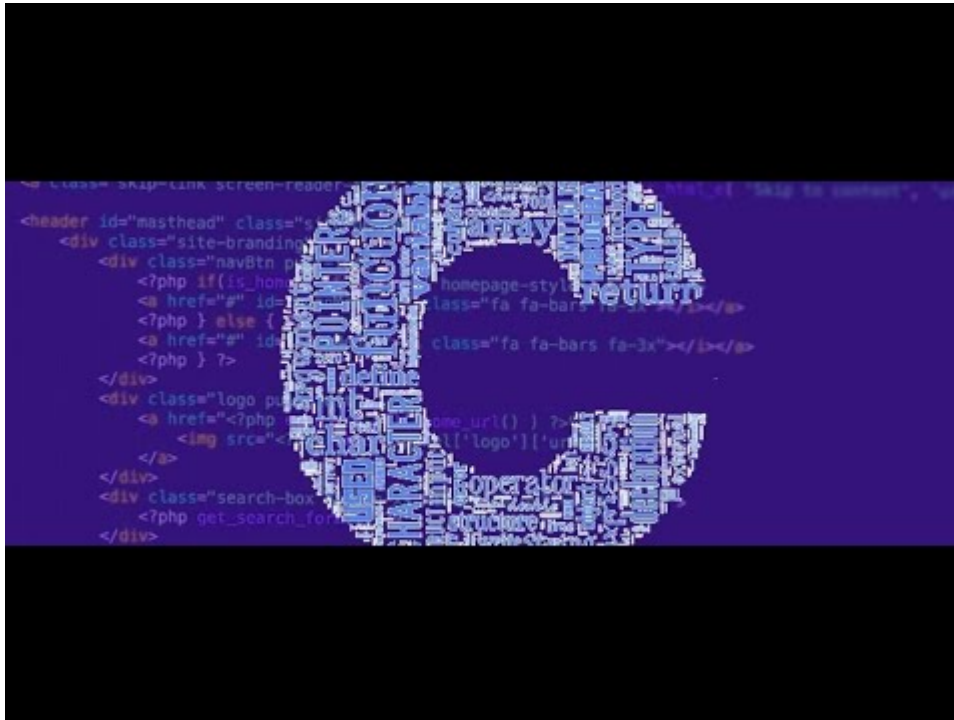
FinFuncion

```

## Diagrama de flujo



## Ejecución



[Suma\\_de\\_cuadrados link](#)

## Funcion **coordenadas\_convert**

text...

Código de la funcion \*\*\*\*:

```
void coordenadas_convert(){
    printf("\n          Conversion de coordenadas          ");
    printf("\n-----");
    printf("\n");
    printf("\n1-Polares a cartesianas");
    printf("\n2-Cartesianas a polares");
    printf("\nIngresa una opción:\t");
    scanf("%d",&opc4);

    switch (opc4)
    {
    case 1:
        polar_to_cartesian();
        break;
    case 2:
        cartesian_to_polar();
        break;
    default:
        printf("Opción invalida");
        break;
    }
}
```

```

}

void polar_to_cartesian(){

    double val = PI / 180;
    double theta,r,x,y;

    printf("Ingrese valor de magnitud(r):  \t");
    scanf("%lf",&r);
    printf("Ingrese valor de angulo(grados):\t");
    scanf("%lf",&theta);

    x=r*cos(theta*val);
    y=r*sin(theta*val);
    printf("\nSalida de coordenadas cartesianas:");
    printf("\n-----");
    printf("\n( x = %0.3lf , y = %0.3lf )\n",x,y);
}

void cartesian_to_polar(){

    double theta=0,r=0,x=0,y=0;
    double val = PI / 180;
    printf("\nIngrese valor de x:\t");
    scanf("%lf",&x);
    printf("Ingrese valor de y:\t");
    scanf("%lf",&y);

    r=sqrt(x*x+y*y);
    theta=atan(y/x);
    theta/=val;

    printf("\nSalida de coordenadas polares:");
    printf("\n-----\n");
    printf("r =  \t %0.3lf \ntheta =\t %0.3lf\n",r,theta);
}

```

## Pseudocodigo

```

Funcion coordenadas_convert
    Escribir '1-Polares a cartesianas'
    Escribir '1-2-Cartesianas a polares'
    Escribir 'ingresa opcion'
    segun opc Hacer
        1:
            polar_to_cartesian
        2:
            cartesian_to_polar

```



```

    FinSegun
FinFuncion

Funcion polar_to_cartesian
    definir val, theta,r,x,y_ como numero
    val=3.1416/180
    Escribir 'Ingresa magnitud'
    Leer r
    Escribir 'Ingresa angulo'
    Leer theta
    x=r*cos(theta*val)
    y_=r*sen(theta*val)

    Escribir x
    Escribir y_

FinFuncion

Funcion cartesian_to_polar
    definir val, theta,r,x,y_ como numero
    val=3.1416/180

    Escribir 'Ingresa x'
    Leer z
    Escribir 'Ingresa y'
    Leer y_

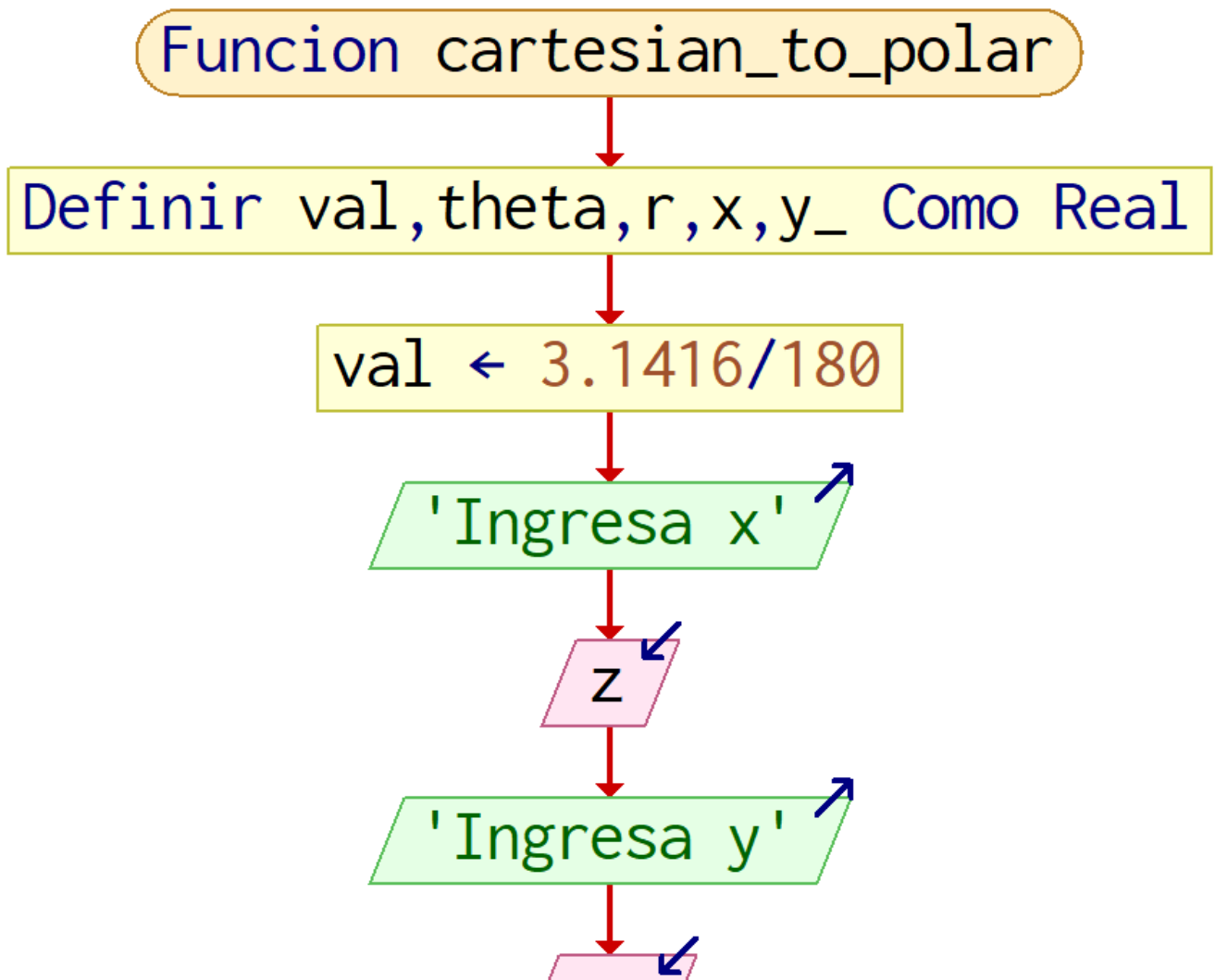
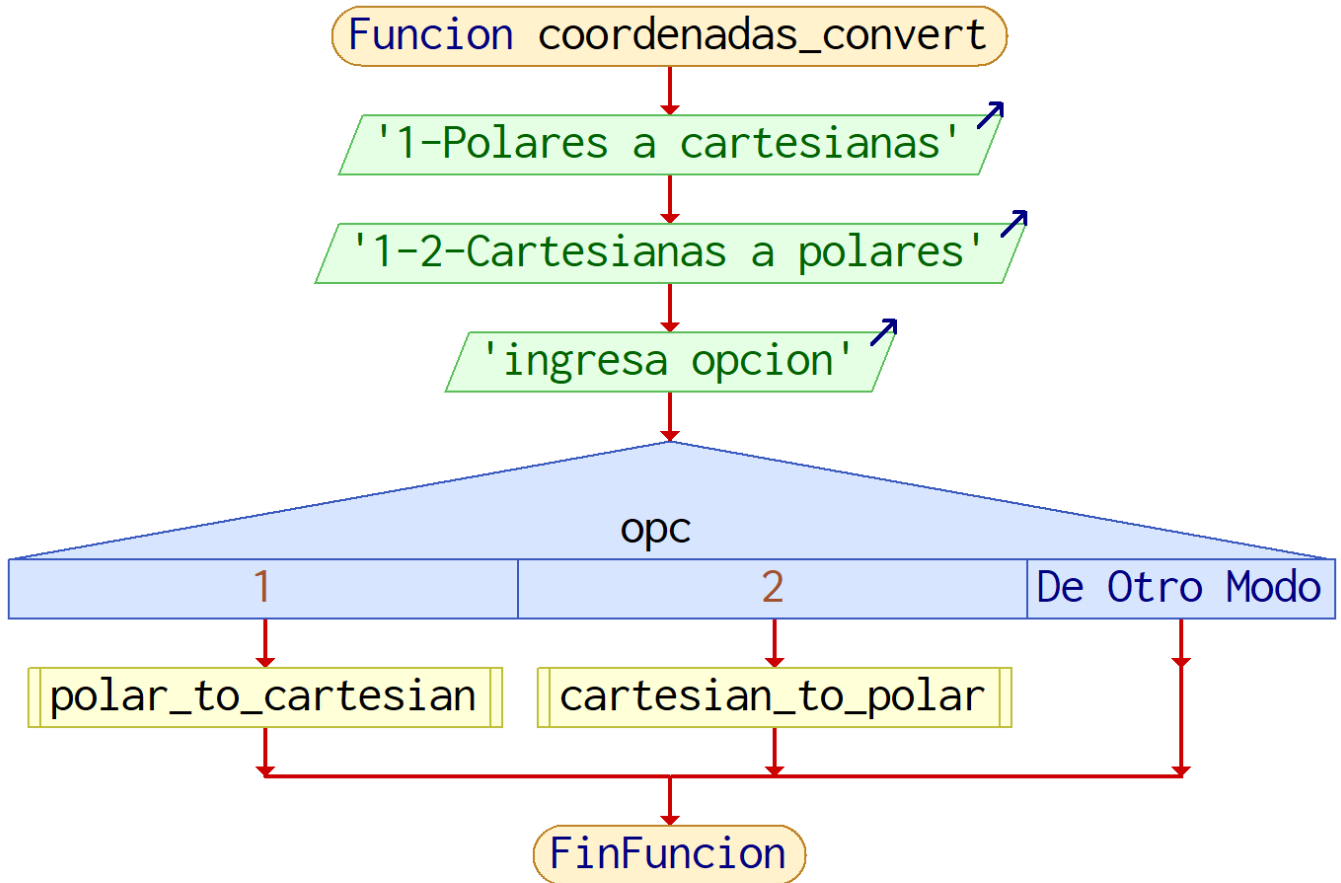
    r=raiz(x*x+y_*y_)
    theta=atan(y_/x)

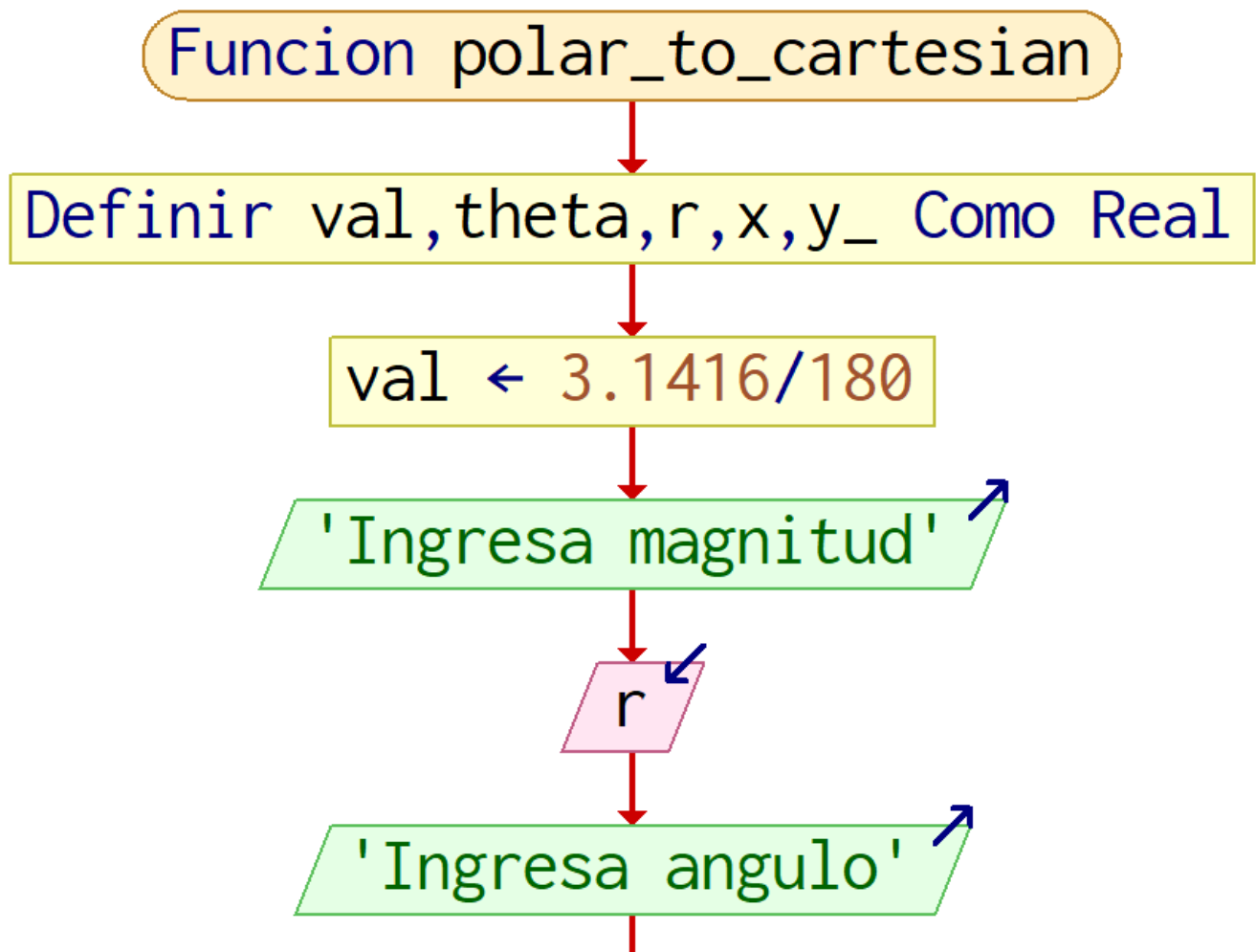
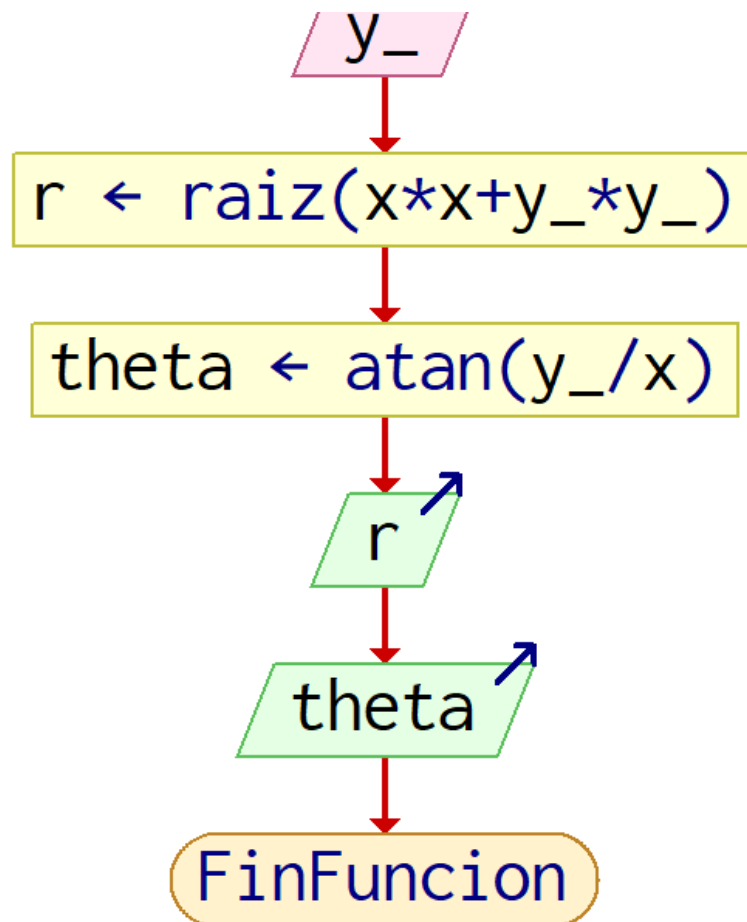
    Escribir r
    Escribir theta

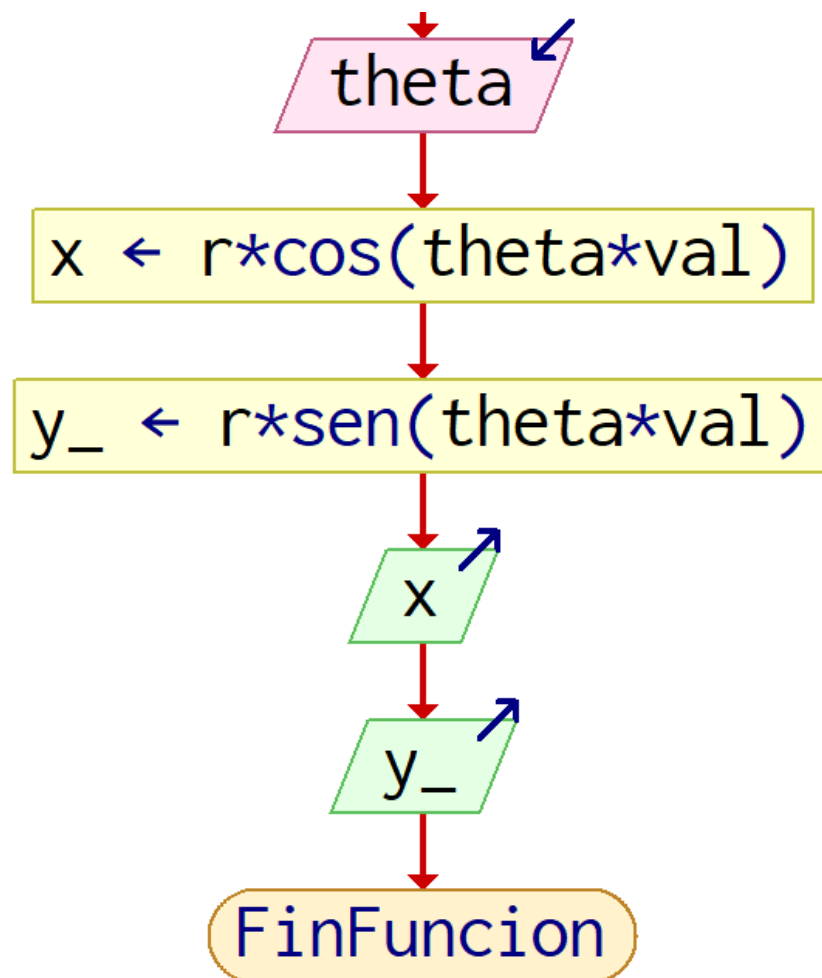
FinFuncion

```

Diagrama de flujo







## Ejecución



[coordenadas\\_convert link](#)

## Funcion **numero\_secreto()**

text...

Código de la funcion **numero\_secreto**:

```
void numero_secreto(){
    int secreto =0;
    int n;
    secreto = rand() % 10 + 1;
    printf("==Descubre el numero entero secreto===== \n\n");
    printf("--El numero secreto esta en el rango de 0 a 9--\t");
    scanf("%d", &n);
    while (secreto != n){
        if(secreto > n) printf("--El numero secreto es mayor--\t");
        else printf("--El numero secreto es menor--\t");

        if (n %2 == 0 && secreto % 2 == 1 ) printf("El numero secreto es
impar--\t");

        if (n % 2 ==1 && secreto % 2 == 0 ) printf("El numero secreto es par--
\t");

        scanf("%d", &n);

    };

    printf("\n * El numero secreto es: %d *", n);
}
```

## Pseudocodigo

```
Funcion numero_secreto
    definir secreto, n Como numero
    secreto=azar(10)
    Escribir 'ingresa el numero secreto'
    Leer secreto
    Mientras secreto <>n Hacer
        si secreto>= n
            Escribir 'el numero es mayor'
        SiNo
            Escribir 'el numero es menor'
        FinSi

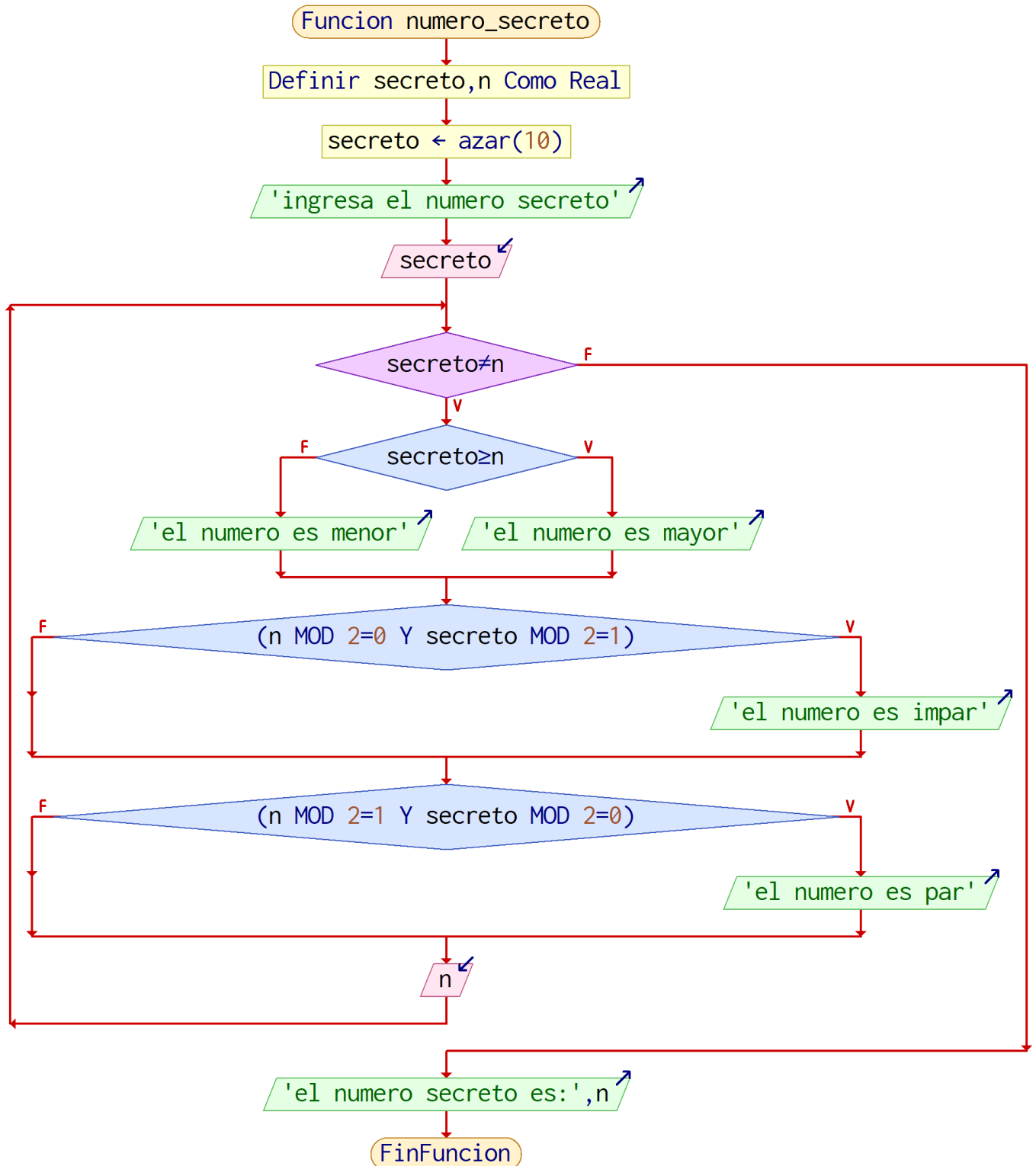
        si(n mod 2=0 y secreto mod 2 =1)
            Escribir 'el numero es impar'
        FinSi

        si(n mod 2=1 y secreto mod 2 =0)
            Escribir 'el numero es par'
```

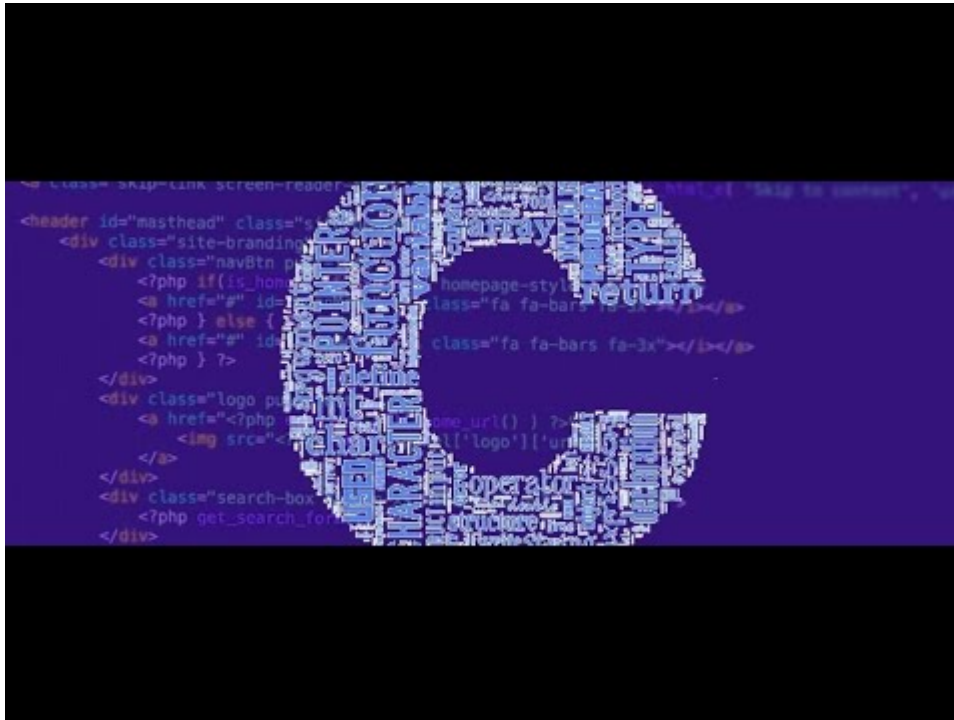
FinSi  
Leer n  
FinMientras

Escribir 'el numero secreto es:',n  
FinFuncion

## Diagrama de flujo



## Ejecución



[numero\\_secreto link](#)

## Funcion **area\_perimetro()**

text...

Código de la funcion **area\_perimetro()**:

```
void area_perimetro(){
    float area, perimetro, a, b;
    printf("Area y perimetro de rectangulo\n");
    printf("-----");
    printf("\nIngresa base:\t");
    scanf("%a",&a);
    printf("Ingresa altura:\t");
    scanf("%a",&b);
    //datos de entrada/

    //lectura de datos/
    //scanf ("%f %f", &a, &b);
    //validacion/
    if(a>0 && b>0 && a<=100 && b<=200){
    //Calculos/
        area= a*b;
        perimetro= 2*a + 2*b;
    //desplegar resultados/
        printf("\nArea = %f,Periemetro = %f\n",area,perimetro);
    } else
```

```

    printf("Las dimensiones del rectangulo no son validas");
}

```

## Pseudocodigo

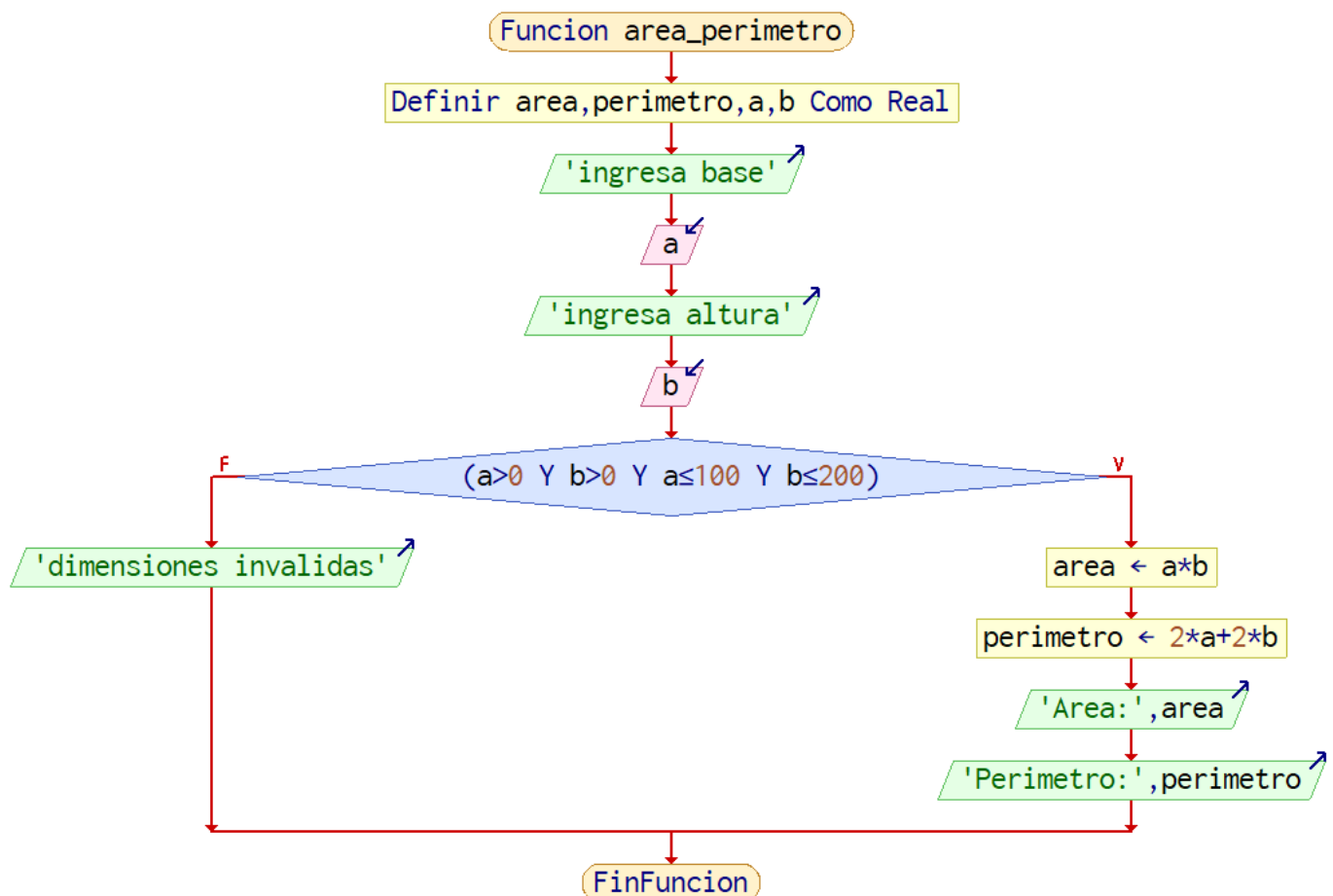
```

Funcion area_perimetro
    Definir area, perimetro, a, b Como Real
    Escribir 'ingresa base'
    Leer a
    Escribir 'ingresa altura'
    leer b

    si(a>0 y b>0 y a<=100 y b<=200)
        area=a*b
        perimetro=2 * a + 2 * b
        Escribir 'Area:',area
        Escribir 'Perimetro:',perimetro
    SiNo
        Escribir 'dimensiones invalidas'
    FinSi
FinFuncion

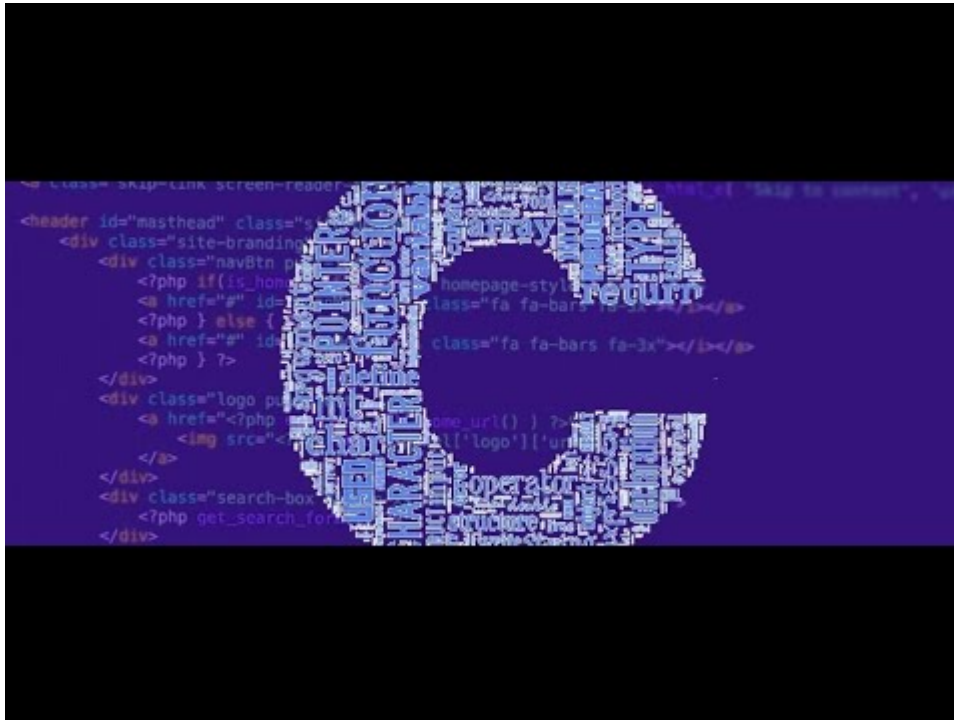
```

## Diagrama de flujo





## Ejecución



[area\\_perimetro link](#)

## Funcion **raices\_ecuacion()**

text...

Código de la funcion **raices\_ecuacion()**:

```
void raices_ecuacion(){

    printf("Raices de una ecuación cuadrática\n");
    printf("Ingresa en orden a b c");
    /* datos de entrada*/
    float a, b, c, d, x1, x2;
    /* leer datos*/
    printf("\n \t\t");
    scanf("%f %f %f", &a, &b, &c);
    //validaciones/
    d= b*b - 4*a*c;

    if(a!=0 && d >= 0){
        x1 =(-b + sqrt(d))/(2*a);
        x2 =(-b - sqrt(d))/(2*a);
        printf("X1 = %f, X2 = %f", x1, x2);
    }else{
        printf("Sin solución");
    }
}
```

```
}

```

## Pseudocodigo

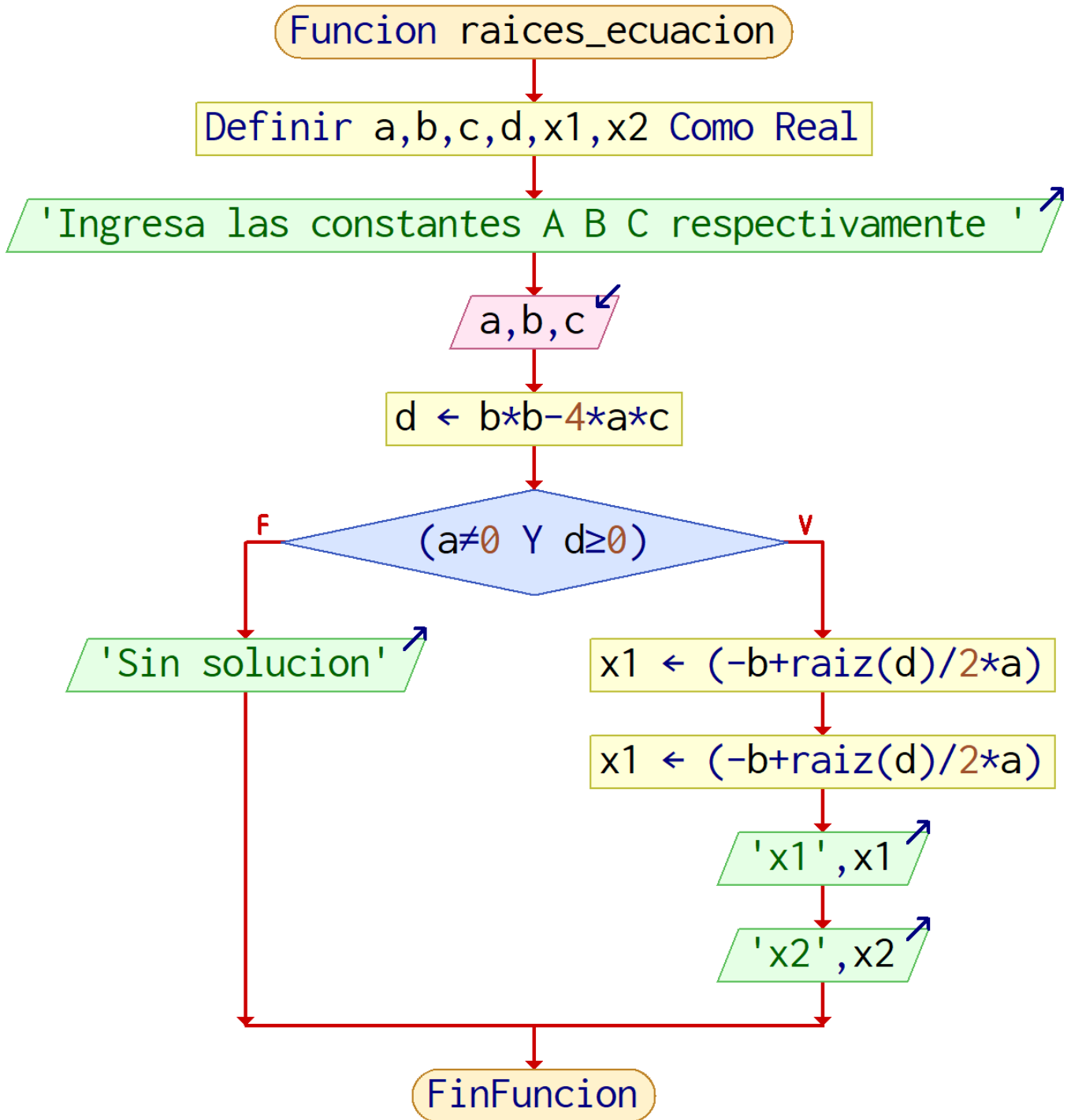
```

Funcion raices_ecuacion
    Definir a,b,c,d,x1,x2 Como numero
    Escribir 'Ingresa las constantes A B C respectivamente '
    leer a,b,c
     $d = b*b - 4*a*c$ 
    si( $a \neq 0$  y  $d \geq 0$ )
         $x1 = (-b + \text{raiz}(d)) / 2*a$ 
         $x1 = (-b + \text{raiz}(d)) / 2*a$ 
        Escribir 'x1',x1
        Escribir 'x2',x2
    SiNo
        Escribir 'Sin solucion'
    FinSi

FinFuncion

```

## Diagrama de flujo



Ejecución



Raices\_ecuacion link

## Funcion **numero\_letras()**

text...

Código de la funcion **numero\_letras()**:

```
void numero_letras(){
    //datos de entrada/
    int numero;
    //lectura de datos/
    printf(" Introduzca un numero entero [1-9]:\t");
    scanf ("%d", &numero);

    //validacion/
    if( numero >=1 && numero <=10)

        switch (numero)
        {
            //desplegar resultados/
            case 1: printf("\n uno");
                    break;
            case 2: printf("\n dos");
                    break;
            case 3: printf("\n tres");
                    break;
            case 4: printf("\n cuatro");
                    break;
            case 5: printf("\n cinco");
                    break;
        }
    }
}
```

```

        case 6: printf("\n seis");
                break;
        case 7: printf("\n siete");
                break;
        case 8: printf("\n ocho");
                break;
        case 9: printf("\n nueve");
                break;
        case 10: printf("\n diez");
                break;
    } else
        printf("\nEL NUMERO DEBE ESTAR EN EL RANGO VALIDO\n");
}

```

## Pseudocodigo

```

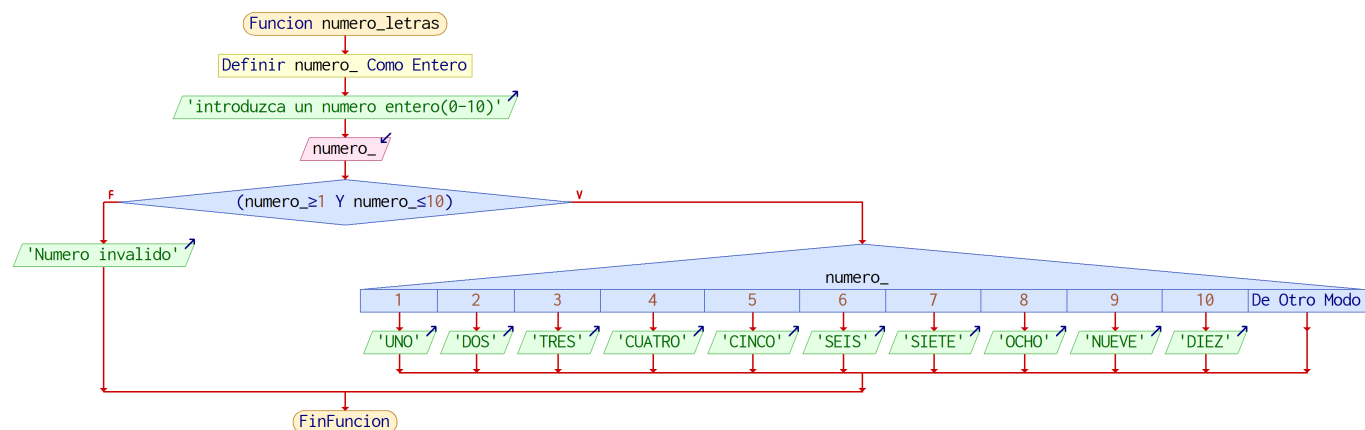
Funcion numero_letras
    definir numero_ Como Entero
    Escribir 'introduzca un numero entero(0-10)'
    leer numero_
    si (numero_ >= 1 y numero_ <= 10)

        segun numero_ hacer
            1:
                Escribir 'UNO'
            2:
                Escribir 'DOS'
            3:
                Escribir 'TRES'
            4:
                Escribir 'CUATRO'
            5:
                Escribir 'CINCO'
            6:
                Escribir 'SEIS'
            7:
                Escribir 'SIETE'
            8:
                Escribir 'OCHO'
            9:
                Escribir 'NUEVE'
            10:
                Escribir 'DIEZ'
        FinSegun
    sino
        Escribir 'Numero invalido'

    FinSi
FinFuncion

```

## Diagrama de flujo



## Ejecución



numero\_letras link