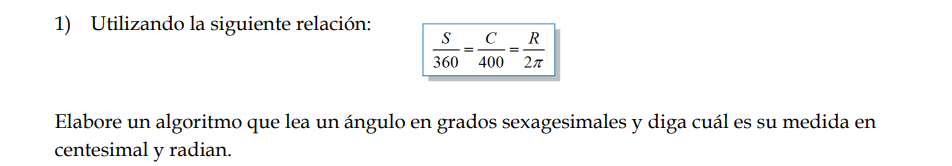
Primer ejercicio



|  |  |  |  |
| --- | --- | --- | --- |
| **ENTRADA** | **PROCESO** | | **SALIDA** |
| **grados** | **formula** | **aplicado** | **resultado** |
| Gsexag | S/360 | S/360 | 120 |
| Gcent | C/400 | Gsexag\*200)/180 | 133 |
| Gradian | R/2PI | Gsexag\*pi)/180);{ | 2 |

* 1. **Modelo**



* 1. **Especificaciones del algoritmo** 
     1. **Diccionario de variables**

**Variables de entrada**

Gsexag, Gcent , Gradian {Representa el valor en grados sexagensimal,centesimal y radian}: tipo Entero

**Variables de proceso**

Gcent=trunc((Gsexag\*200)/180); {Representa el valor en grados centesimales }: tipo Entero

Gradian=trunc((Gsexag\*pi)/180);{Representa el valor en grados radianes}: tipo Entero

**Variables de salida**

"n/En centesimales es; ",Gcent; {Representa el valor en grado centesimal }:tipo Entero

"En radian es; ", Gradian;{Representa el valor en grado radian }:tipo Entero

* + 1. **Pre condición**

trunc((Gsexag\*pi)/180)

* + 1. **Acción del algoritmo**

**Calcular en grado sexagesimal, centesimal y radian**

* + 1. **Post condición**

( Gsexag, Gcent , Gradian son de tipo entero }

1. **Diseño**
   1. **Descripción del algoritmo**

**Primera fase**

Inicio { Sexagecimal\_Medida\_Cent\_y\_Radian }

{Declaración e inicialización de variables}

{Leer dato}

{ "Digite en grados sexagesimales "}

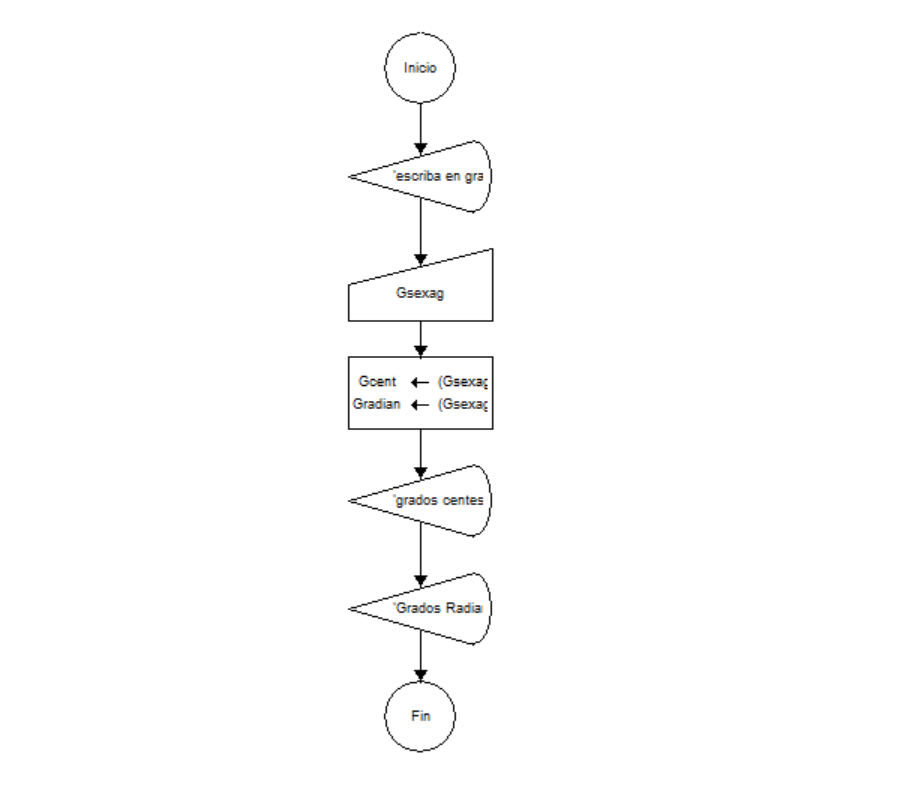
{Escribir "n/En centesimales es; "}

{ Escribir "En radian es; "}

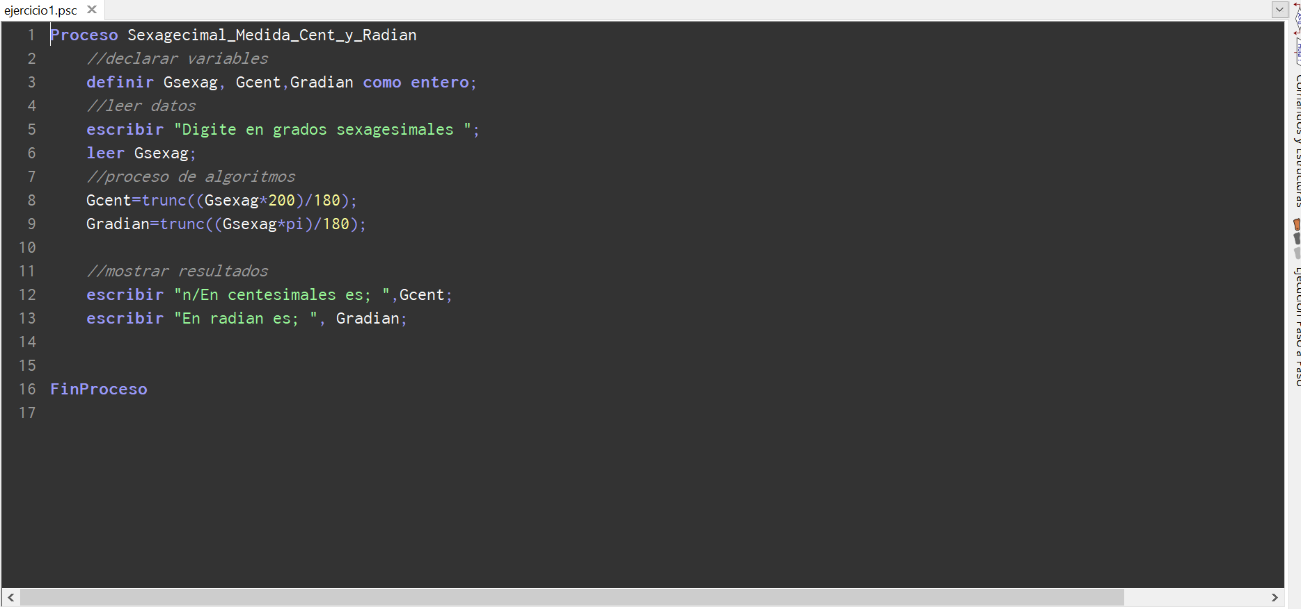
Fin

**Segunda fase**

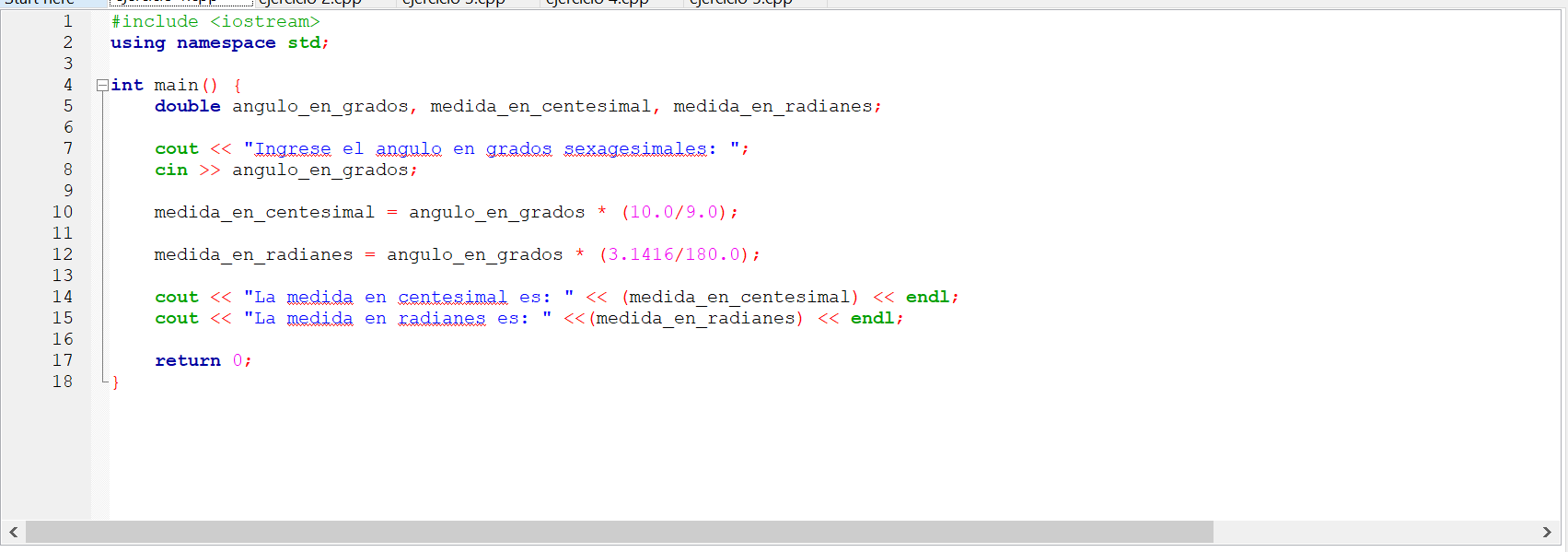
Diseño del diagrama de flujo



* 1. **Definición del algoritmo**



Definición en c++



Segundo ejercicio



|  |  |  |  |
| --- | --- | --- | --- |
| **ENTRADA** | **PROCESO** | | **SALIDA** |
| **numeros** | **formula** | **aplicado** | **resultado** |
| nPerson | am=(nPerson\*(nPerson-1))/2; | 12 | 66 |

* 1. **Modelo**

**am=(nPerson\*(nPerson-1))/2;**

* 1. **Especificaciones del algoritmo** 
     1. **Diccionario de variables**

**Variables de entrada**

nPerson, am {Representa el número de personas de una reunión a ingresar al programa}: tipo Entero

**Variables de proceso**

NPerson {Representa el numero de personas }: tipo Entero

Calcular {Representa el calcular de apretones de manos }: tipo Entero

**Variables de salida**

Calcular {Representa el calcular de apretones de manos }: tipo Entero

* + 1. **Pre condición**

{ am=(nPerson\*(nPerson-1))/2; }

* + 1. **Acción del algoritmo**

**Calcular cuantos apretones de manos en aquella reunion**

* + 1. **Post condición**

{am,nPerson son de tipo entero positivo}

i

1. **Diseño**
   1. **Descripción del algoritmo**

**Primera fase**

Inicio { Apretones\_de\_mano }

{Declaración e inicialización de variables}

{Leer dato}

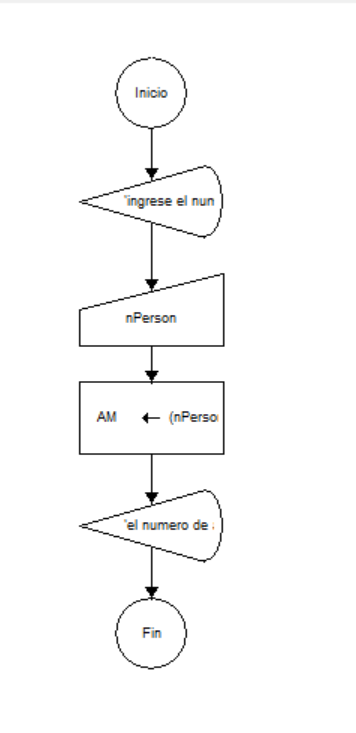
{Calcular los apretones de manos de aquella reunion}

{Escribir el números de personas es:” ,am

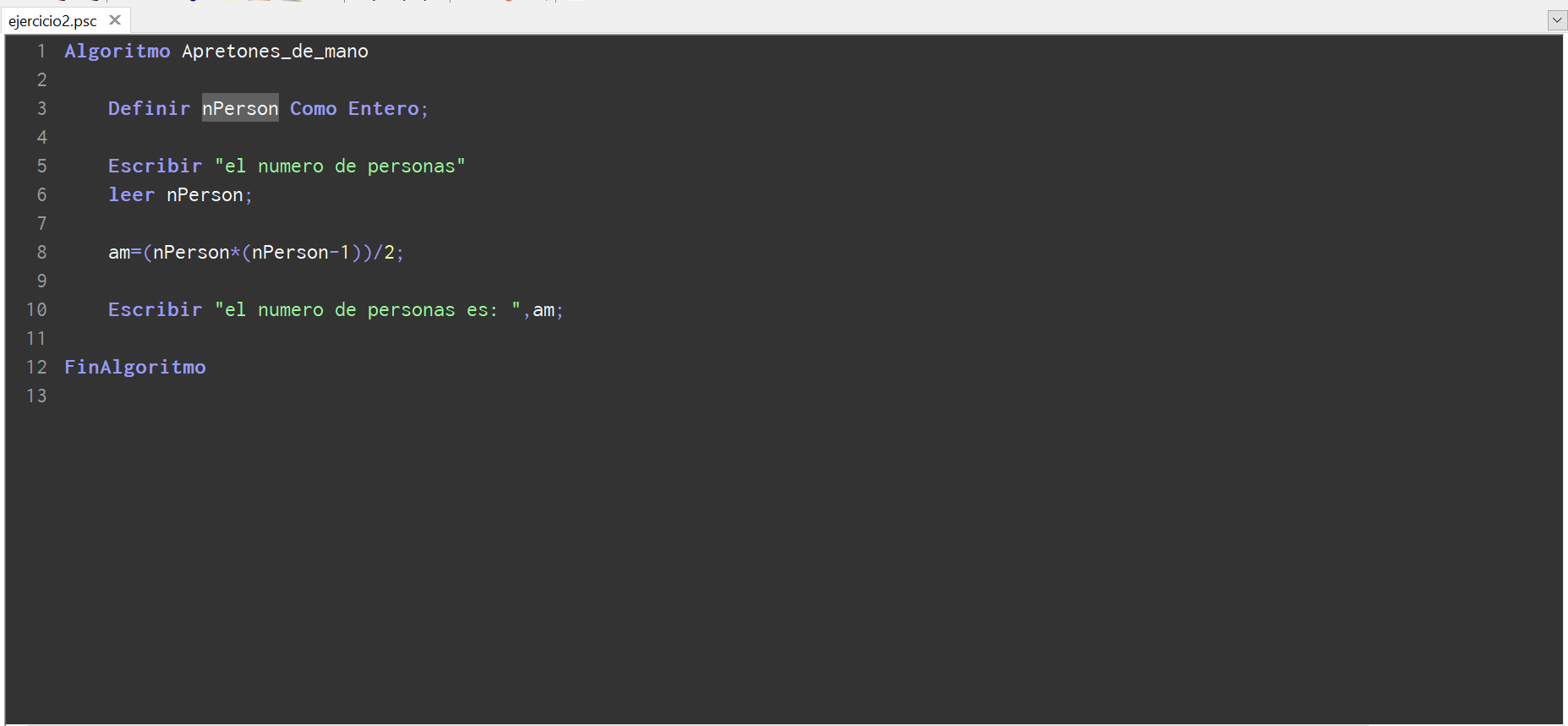
Fin

**Segunda fase**

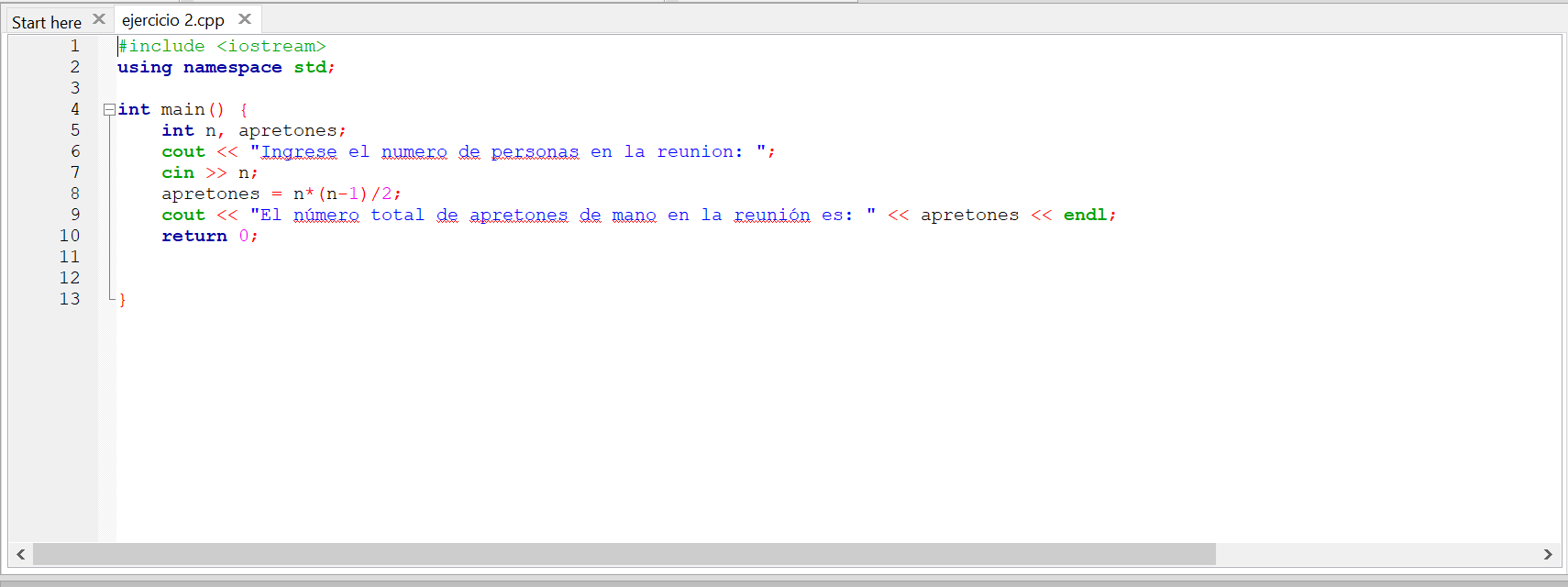
Diseño del diagrama de flujo



* 1. **Definición del algoritmo**



* 1. **Definición del algoritmo c++**



**TERCER EJERCICIO**

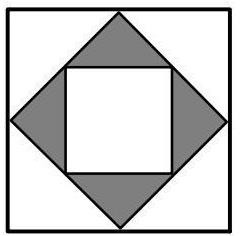
**Descripción del problema:**

En un cuadrado cuyo lado es a, se unen los puntos medios de sus cuatro lados, formándose

otro cuadrado cuyos puntos medios se unen también formando otro cuadrado, y así

sucesivamente. Calcule la suma de las áreas de todos los cuadrados así formados.

**a**



**a**

**Definición de la solución:**

1. **Análisis**
   1. **Entender el problema**

|  |  |  |  |
| --- | --- | --- | --- |
| **ENTRADA** | **PROCESO** | | **SALIDA** |
| **lado** | **area** | **Sumatoria** | **sumatoria** |
| 20 | (20) ² | 2(400) ² | 320000 |
| 89 | (89) ² | 2(7921) ² | 125484482 |
| 51 | (51) ² | 2(2601) ² | 13530402 |

* 1. **Modelo**

area=(lado) ²

sumatoria=2(area) ²

* 1. **Especificaciones del algoritmo** 
     1. **Diccionario de variables**

**Variables de entrada**

lado {Representa el valor del lado del cuadrado a ingresar al programa}: tipo Entero

**Variables de proceso**

area {Representa el valor del area del primer cuadrado}: tipo Entero

sumatoria {Representa la suma de todas las áreas de los cuadrados}: tipo Entero

**Variables de salida**

sumatoria {Representa la suma de todas las áreas de los cuadrados}: tipo Entero

* + 1. **Pre condición**

{| lado > 0 |}

* + 1. **Acción del algoritmo**

Calcular la suma de todas las áreas de los cuadrados

* + 1. **Post condición**

{arae, sumatoria son de tipo entero positivo}

1. **Diseño**
   1. **Descripción del algoritmo**

**Primera fase**

Inicio {hallar la suma de las áreas de los cuadrados}

{Declaración e inicialización de variables}

{Leer dato}

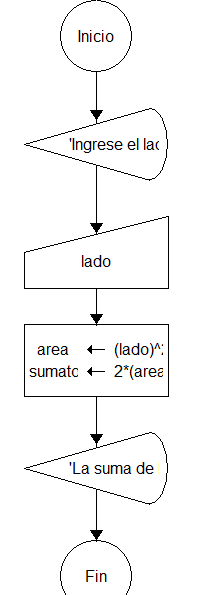
{Calcular la suma de las áreas de los cuadrados}

{Escribir el resultado de las áreas sumadas}

Fin

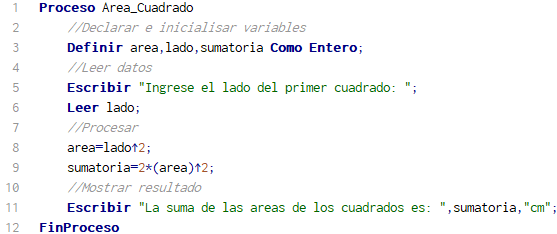
**Segunda fase**

Diseño del diagrama de flujo



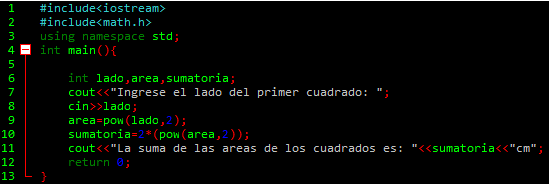
Diseño del diagrama de flujo

* 1. **Definición del algoritmo**



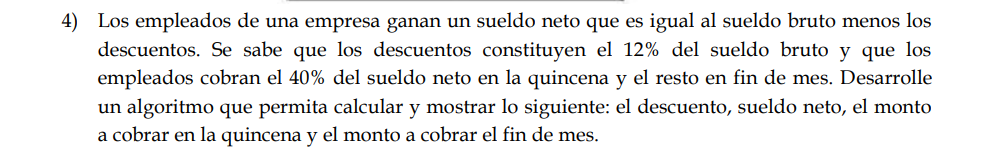
Diseño de PSeInt

1. **Codificación en C++**

****

Algoritmo de C++

Cuarto ejercicio



* 1. **Modelo**

**tipomovil**

* 1. **Especificaciones del algoritmo** 
     1. **Diccionario de variables**

**Variables de entrada**

saldoant, montotran, saldoactual {Representa el valor del saldo a ingresar al programa}: tipo real

tipomovil Como Carácter

**Variables de proceso**

tipomovil="R" {Representa el valor del retiro }: tipo real

tipomovil="D" {Representa el valor del depósito }: tipo Entero

**Variables de salida**

Escribir "su saldo actual es de: ", saldoactual {Representa la suma de todas las áreas de los cuadrados}: tipo real

* + 1. **Pre condición**

{ saldoactual=saldoant-montotran }

* + 1. **Acción del algoritmo**

Calcular saldo actual **Post condición**

{ tipomovil Como (Carácter) y saldoant, montotran, saldoactual

son de tipo real }

1. **Diseño**
   1. **Descripción del algoritmo**

**Primera fase**

Inicio {hallar el saldo actual }

{Declaración e inicialización de variables}

{Leer dato}

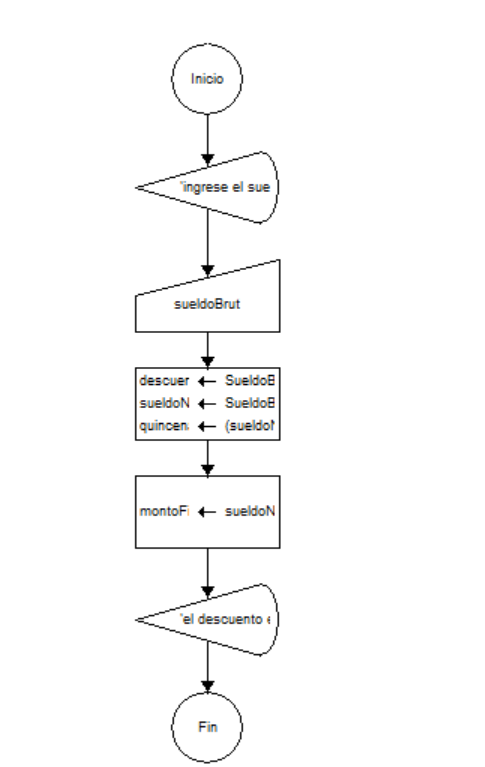
{Calcular el tipo de movimiento retiro (R) o deposito (D)" }

{Escribir el resultado del saldo actual }

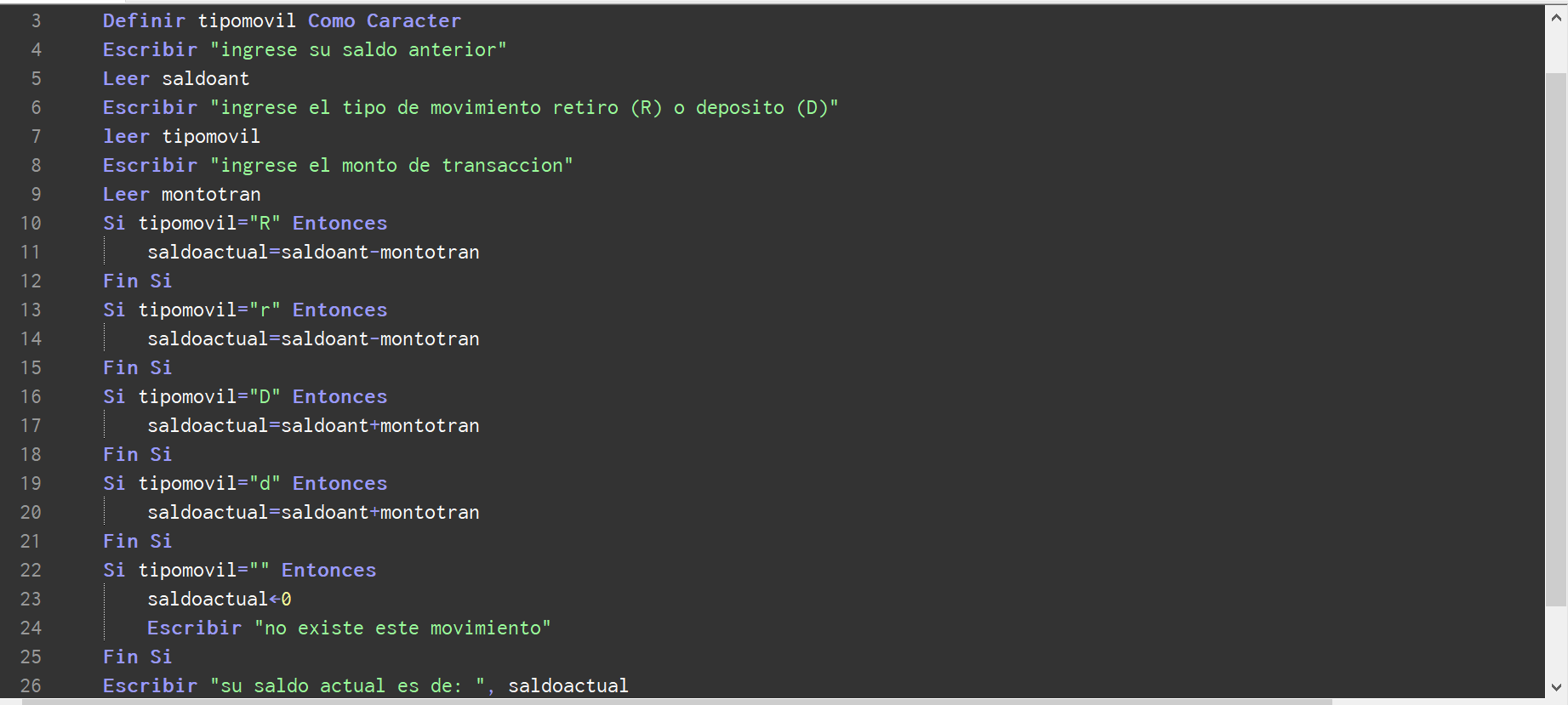
Fin

**Segunda fase**

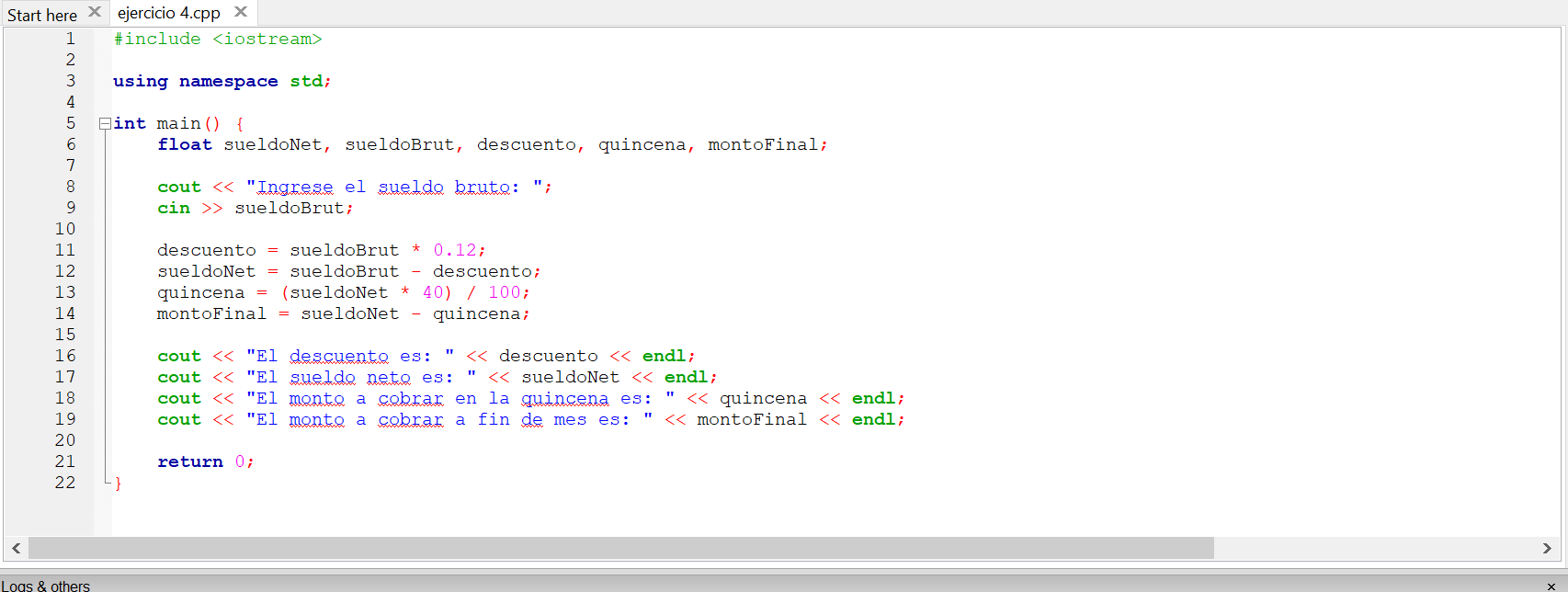
Diseño del diagrama de flujo



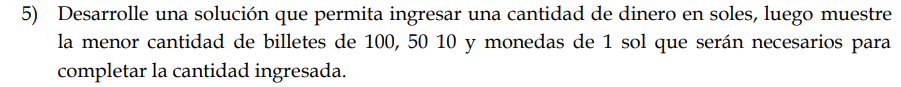
* 1. **Definición del algoritmo**



* 1. **Definición del algoritmo c++**



Quinto ejercicio



* 1. **Modelo**

Billetes de 100 = cantidad\_de\_dinero / 100;

**Especificaciones del algoritmo**

* + 1. **Diccionario de variables**

**Variables de entrada**

Billetes de 100,50,10 y monedas de 1 sol {Representa la cantidad de ssoles a ingresar al programa}: tipo Real

**Variables de proceso**

billetes\_de\_100 / 100 {Representa el valor del billete de 100}: tipo real

resto {Representa el resto}: tipo real

**Variables de salida**

"escribir Billetes de 100: " ,billetes\_de\_100 {Representa la cantidad de billetes de 100 soles }: tipo real

"escribir Billetes de 50: ",billetes\_de\_50 ; {Representa la cantidad de billetes de 50 soles }: tipo real

"escribir Billetes de 10: ",billetes\_de\_10; {Representa la cantidad de billetes de 10 soles }: tipo real

"escribir Monedas de 1 sol: ",monedas\_de\_1; {Representa la cantidad de monedas de 1 sol}: tipo real

* + 1. **Pre condición**

{resto = cantidad\_de\_dinero % 100; }

* + 1. **Acción del algoritmo**

**Calcular la cantidad de dinero en soles que serán necesarios para completar el monto ingresado**

* + 1. **Post condición**

{ cantidad de dinero en soles}

1. **Diseño**
   1. **Descripción del algoritmo**

**Primera fase**

Inicio {ingrese la cantidad de soles }

{Declaración e inicialización de variables}

{Leer dato}

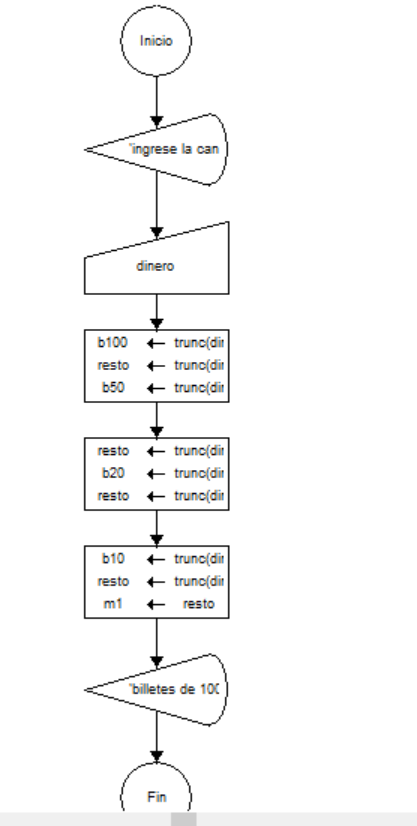
{Calcular cuantos billetes de 100,50,10 y monedas de 1 sol }

{Escribir resultado del calcular la cantidad de billetes de 100,50, 10 y moneda de sol que falta para completar el monto}

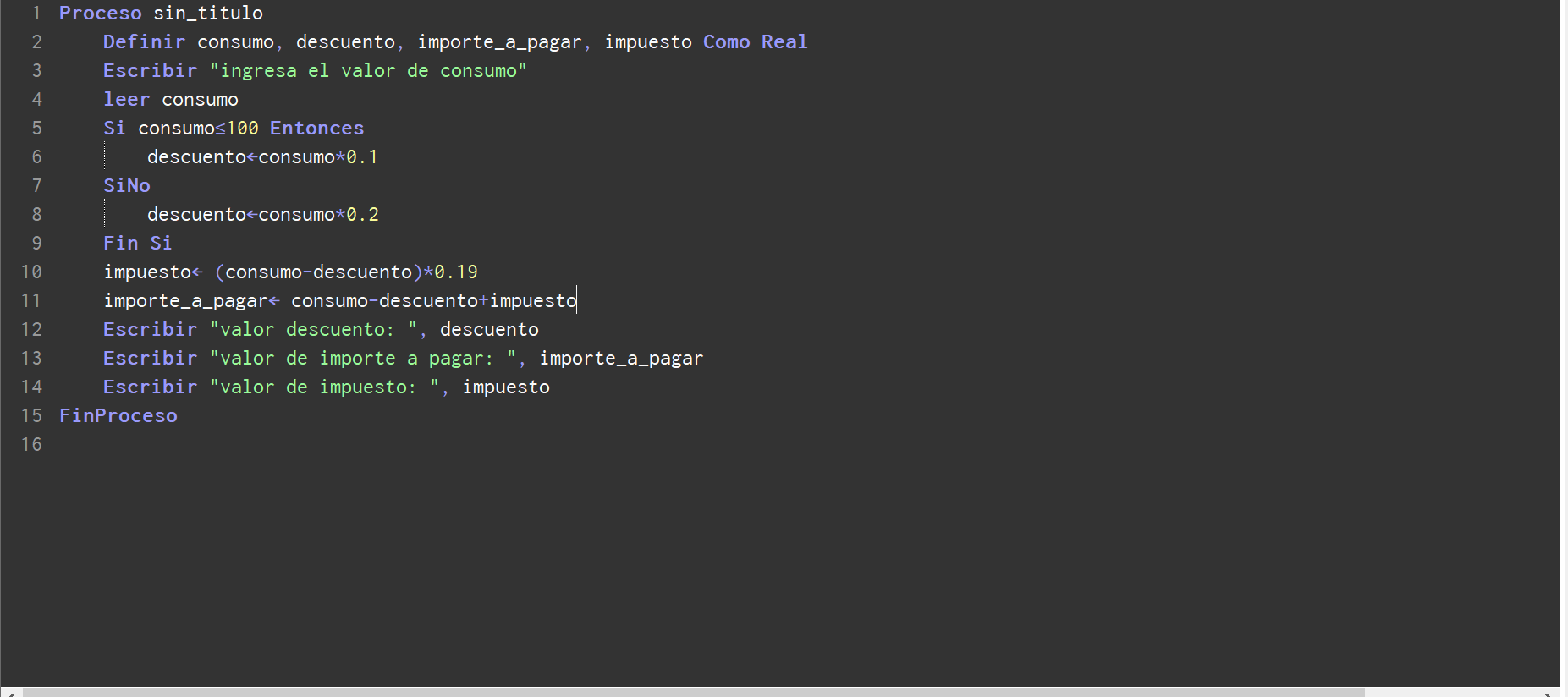
Fin

**Segunda fase**

Diseño del diagrama de flujo



* 1. **Definición del algoritmo**



* 1. **Definición del algoritmo c++**

