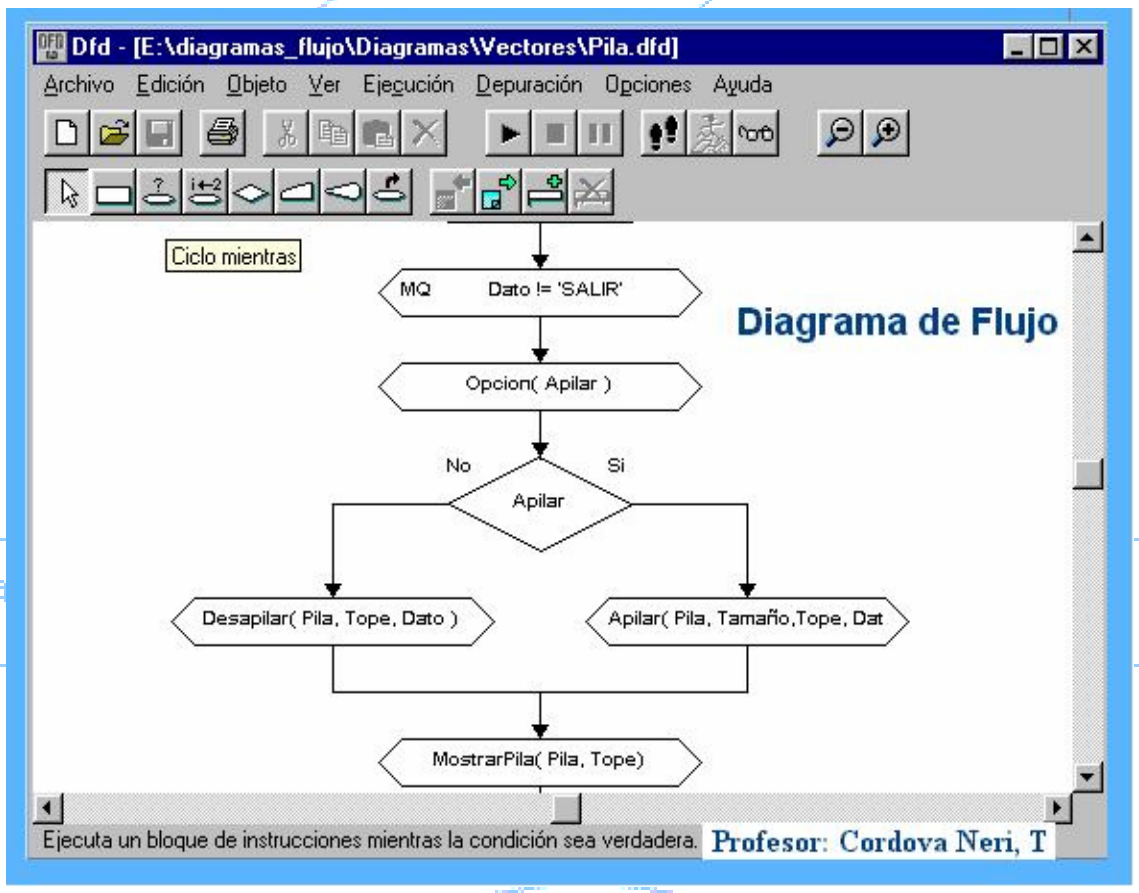
DIAGRAMA DE FLUJO DE DATOS



AUTOR:

CORDOVA NERI, TEODORO

Lima – Perú

INTRODUCCIÓN

La presente guía denominada **DIAGRAMA DE FLUJO DE DATOS,** ilustra una de las técnicas para representar "Soluciones" a problemas del Mundo Real en forma visual, es decir; en forma grafica.

Esta **técnica mediante graficas de Diagrama de Flujo**, ilustra como diseñar los procedimientos o sentencias con coherencia lógica, que representan la solución al problema planteado..

Hasta la presente década, para el desarrollo de cursos, tales como **Algoritmos y Estructuras de Datos,** no ha existido un **Software** que permita implementar el Diagrama de Flujo del problema planteado y que en especial permita su **Ejecución (Compilación) y ver los resultados dentro del mismo diagrama de flujo, según el objetivo del problema.** Es decir; Ud. puede comprobar la lógica de su algoritmo, sin utilizar algún Compilador Real o Lenguaje de Programación específico (Turbo Pascal, Borland C++ 5.0, etc).

Motivo por el cual, y como Docente responsable de la Asignatura de Lenguajes Algorítmicos por más de una década, presento los problemas y su solución usando el *Software* *(Diagrama de Flujo de Datos),* producto desarrollado en la Universidad del Magdalena Santa Marta, Colombia.

Este producto, cubre en forma eficiente la ejecución de programas usando Estructuras de Control, Vectores, matrices y Programación Modular Dependiente, pero el Software tiene limitaciones para implementar problemas usando Registros, Archivos, Punteros y Diseño de Programación Independiente

Los Programas Fuentes Ud. Puede encontrarlo en las textos de : **Algoritmos** en **Borland Pascal For Windows versión 7.0** o en el texto **Algoritmos y sus Aplicaciones en Borland C++ 5.0.** Obras publicadas por el autor.

Me es grato agradecer las sugerencias de colegas que en su debida oportunidad aportaron con sus opiniones para la mejora de la presente.

El Autor

CAPITULO I

Opciones del Software

Procedimientos

1.- Ejecutar DFD

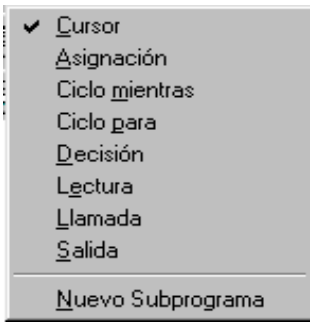
2.- Presentación del Software en Modo Edición:

3.- Opción **Archivo**: Permite crear nuevo archivo, Guardar , imprimir, salir

4.- Opción **Edicion**: Permite copiar, pegar, insertar, eliminar, otros.

5.- Opción **Objeto**: Permite ejecutar las opciones mostradas en el grafico (Graf_a)

(Graf_a)



En esta opción cubre todas las bondades que brinda el Software en mención. Tales Asignación, Estructuras de control: Mientras<cond> , Para , Decisión, etc.

6.- Opción **Ver**: Permite aumentar o disminuir el Diagrama, depurar, etc.

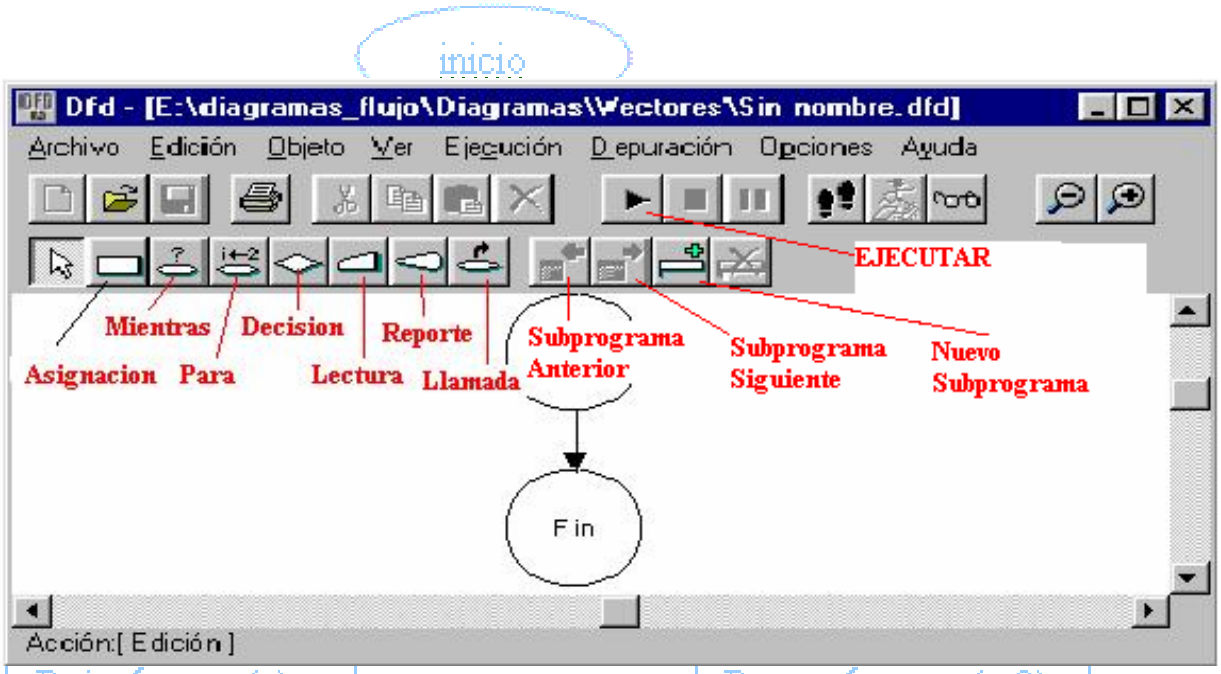
7.- Opción **Ejecución**: Permite ejecutar F9, Pausar

8.- Opción **Depuracion**: Permite ejecutar paso simple F7, evaluar F5

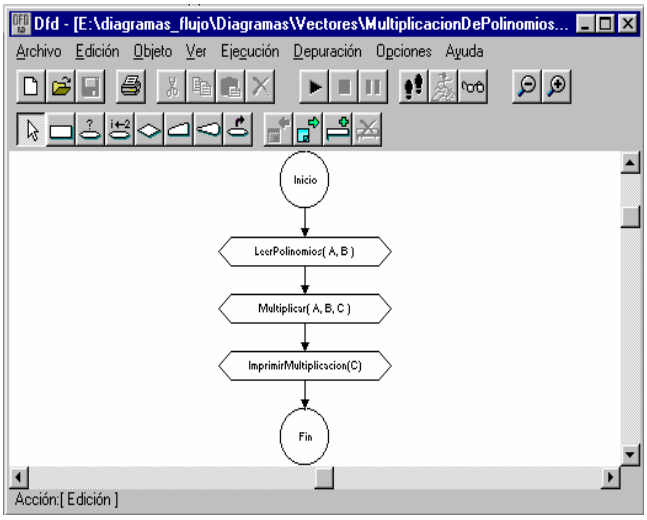
9.- Opción **Opciones**: Permite usar ángulos en grados, radianes.

10.- Opción **Ayuda**: Brinda ayuda al lector

11.-Descripción de algunos botones.



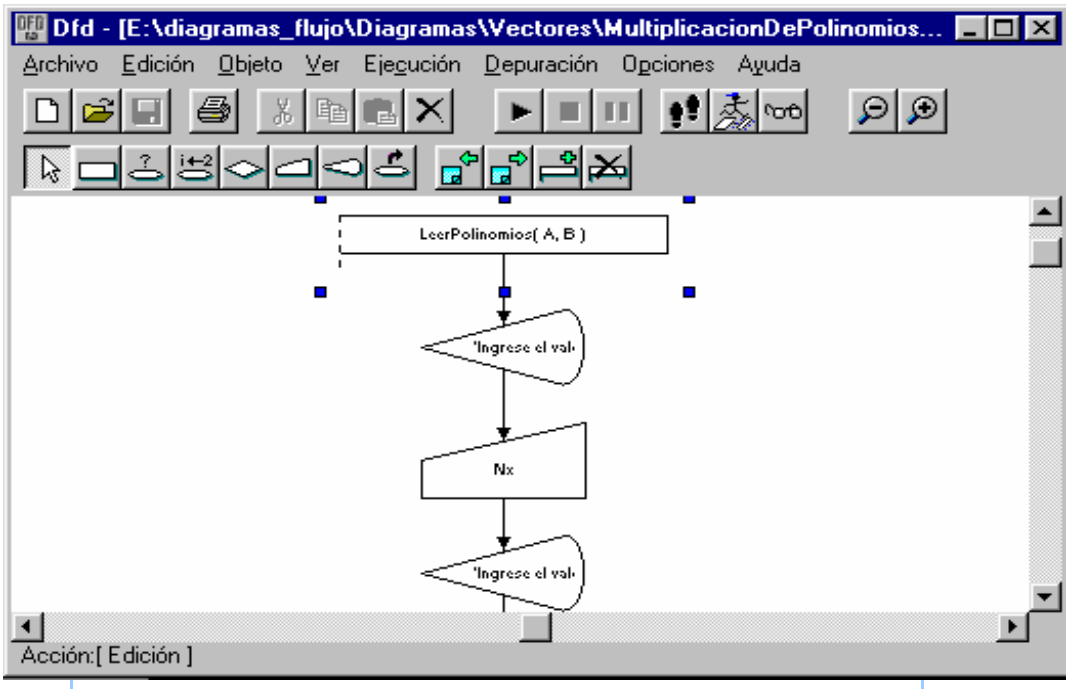
12.- Los botones: **Subprograma Anterior** y **Subprograma Siguiente** se activan cuando su diagrama de Flujo tiene Subprogramas. En el siguiente grafico se ilustra la interacción respectiva entre cada subprograma:



En esta grafica, se ilustra el Modulo Principal del Algoritmo para calcular las raíces de un Polinomio de grado n. En el presente diagrama de flujo se ha diseñado 3 subprogramas: LeerPolinomios, Multiplicar, ImprimirMultiplicacion.

La flecha a la derecha, indica que hará una llamada (call) al siguiente subprograma.

En la siguiente grafica, se ilustra el procedimiento **LeerPolinomios()**



En esta grafica, se ilustra el Diseño de Procedimiento LeerPolinomios(). Observara que los botones se han activado con Flecha con dirección Izquierda y Flecha con dirección Derecha indicando que puede salir o ingresar a otro subprograma.

Si el diagrama tiene mas subprogramas Ud. Continua con flecha a derecha hasta llegar al ultimo, en este caso se desactiva indicando que no existen mas subprogramas.

Si desea Eliminar Subprogramas usar el botón

El botón indica **Paso simple**, es decir; ejecutar por pasos (bloques).

El botón indica **Ejecutar Hasta,** significa que puede ejecutar parcialmente el programa hasta donde avanzo.

El botón indica **Depurador.**

CAPITULO II

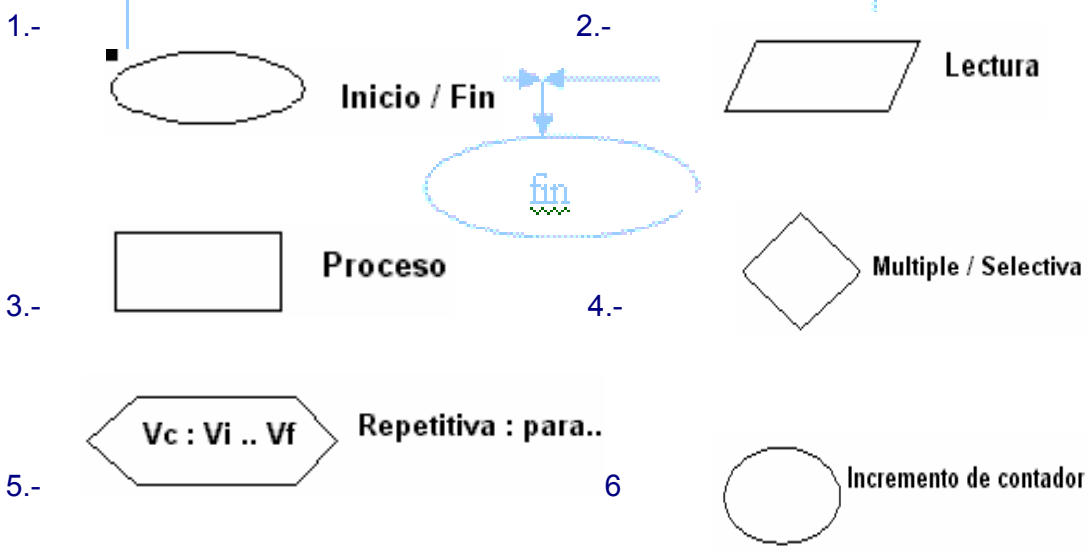
Aplicaciones usando Diagrama de Flujo

Un Diagrama de Flujo de Datos es una descripción gráfica de un procedimiento para la resolución de un problema. Son frecuentemente usados para describir algoritmos y programas de computador. Los diagramas de flujo de datos están compuestos por figuras conectadas con flechas. Para ejecutar un proceso comienza por el INICIO y se siguen las flechas de figura a figura, ejecutándose las acciones indicadas por cada figura; el tipo de figura indica el tipo de paso que representa.

Del Software, DFD es un software diseñado para construir y analizar algoritmos Ud. puede crear diagramas de flujo de datos para la representación de algoritmos de programación estructurada a partir de las herramientas de edición que para éste propósito suministra el programa. Después de haber ingresado el algoritmo representado por el diagrama, podrá ejecutarlo, analizarlo y depurarlo en un entorno interactivo diseñado para éste fin. La interfaz gráfica de DFD, facilita en gran medida el trabajo con diagramas ya que simula la representación estándar de diagramas de flujo en hojas de papel.

Elementos

Para inicia Primero las aplicaciones, primero se definen los elementos de un Diagrama de Flujo:



7.- Lineas de flujo

8.- Imprimir

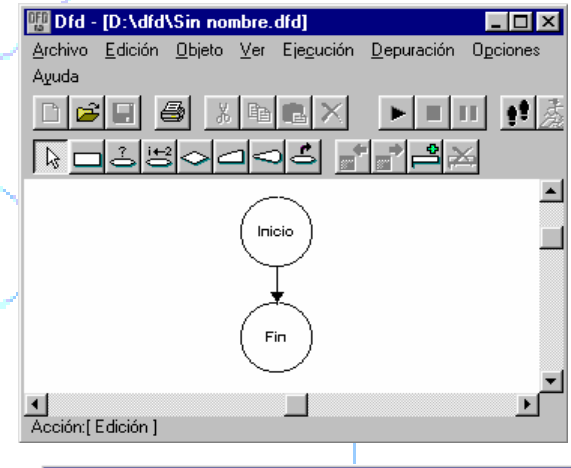
1.- Estructuras Secuenciales

Para diseñar un diagrama de flujo con estas estructuras, se usa los procedimientos de: Lectura, Procesos y Reportes.

Los Diagramas de este tipo, se les denomina Programas Secuenciales o lineales, pues no tiene vuelta a tras(bucles).

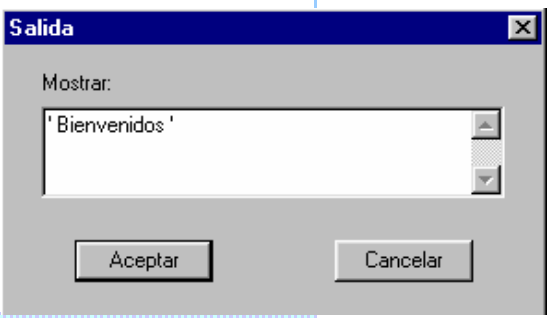
Problema #1

Diseñar un Diagrama de Flujo que calcule la suma de 2 números y genere su reporte respectivo. El diagrama debe solicitar el ingreso de 2 números. La suma se calcula SumaNum=a+b



Paso 1.- Ejecutar DFD. Presenta la pantalla principal de Modo Edición.

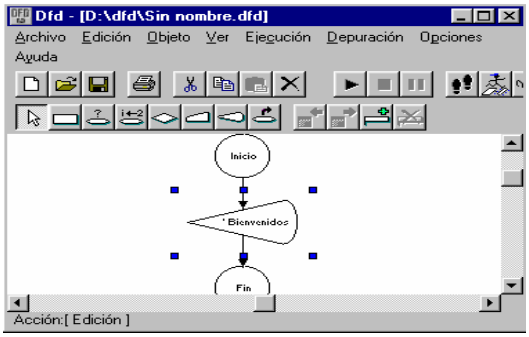
Paso 2.- Inserte el botón de Entrada/salida para enviar un Mensaje al usuario indicando ' Bienvenidos'.



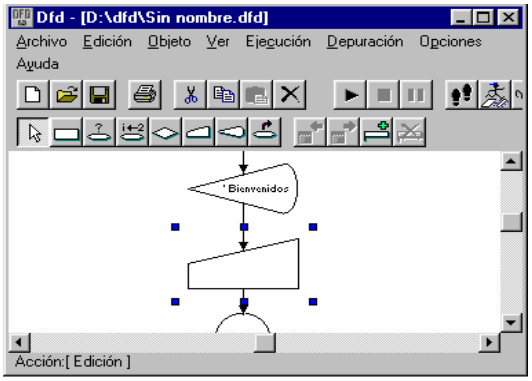
Hacer doble clic en el y luego presenta el siguiente formulario indicando que edite el mensaje:

Observacion. Cuando ingrese caracteres o cadenas, debe usar apostrofe " ' ", al inicio y al final de la cadena.

En nuestro caso, 'Bienvenidos' y presione Aceptar. La inserción se presenta en la siguiente figura:



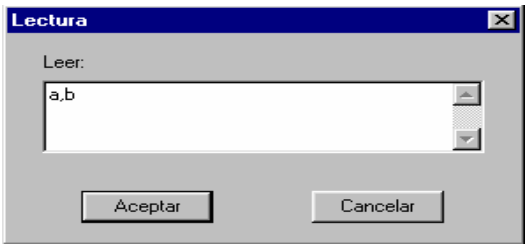
Paso 3.- Lectura de Datos(Ingreso de Datos): Usar el símbolo de Lectura  , inserte después del símbolo que contiene el mensaje de bienvenida.



En la siguiente grafica se ilustra la nueva inserción, el cual se encuentra activado.

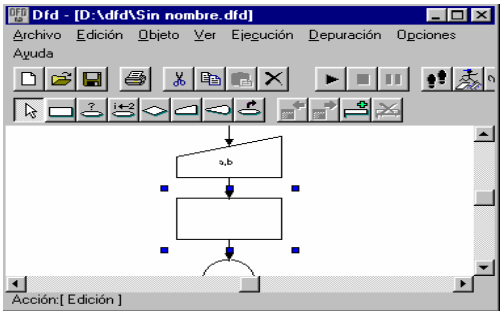
A este símbolo se debe agregar las variables que usa el algoritmo para calcular la suma de los 2 números.

Hacer doble clic en símbolo y presenta el siguiente formulario indicando que edite las variables respectivas.



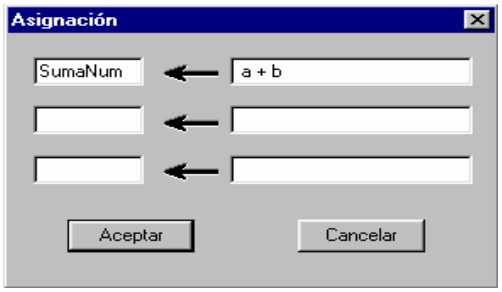
En nuestro problema se considera 2 variables: a, b. Luego Aceptar.

Paso 4.- Proceso de Datos (Transformación de Datos): inserte después del símbolo que contiene la definición de las variables. Para lo cual debe usar el símbolo de Proceso  , aquí debe editar la formula que calcule la suma.

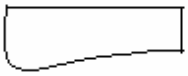


En la siguiente grafica se ilustra la nueva inserción, el cual se encuentra activado.

Ahora hacer doble clic y muestra el siguiente formulario indicando que edite la formula SumaNum=a+b para calcular la suma. Debe presionar el botón Aceptar.

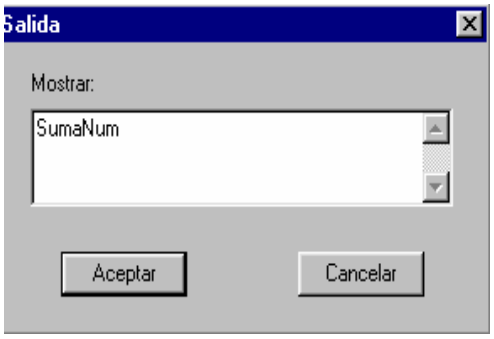


En la siguiente grafica se ilustra el formulario para asignar expresiones.

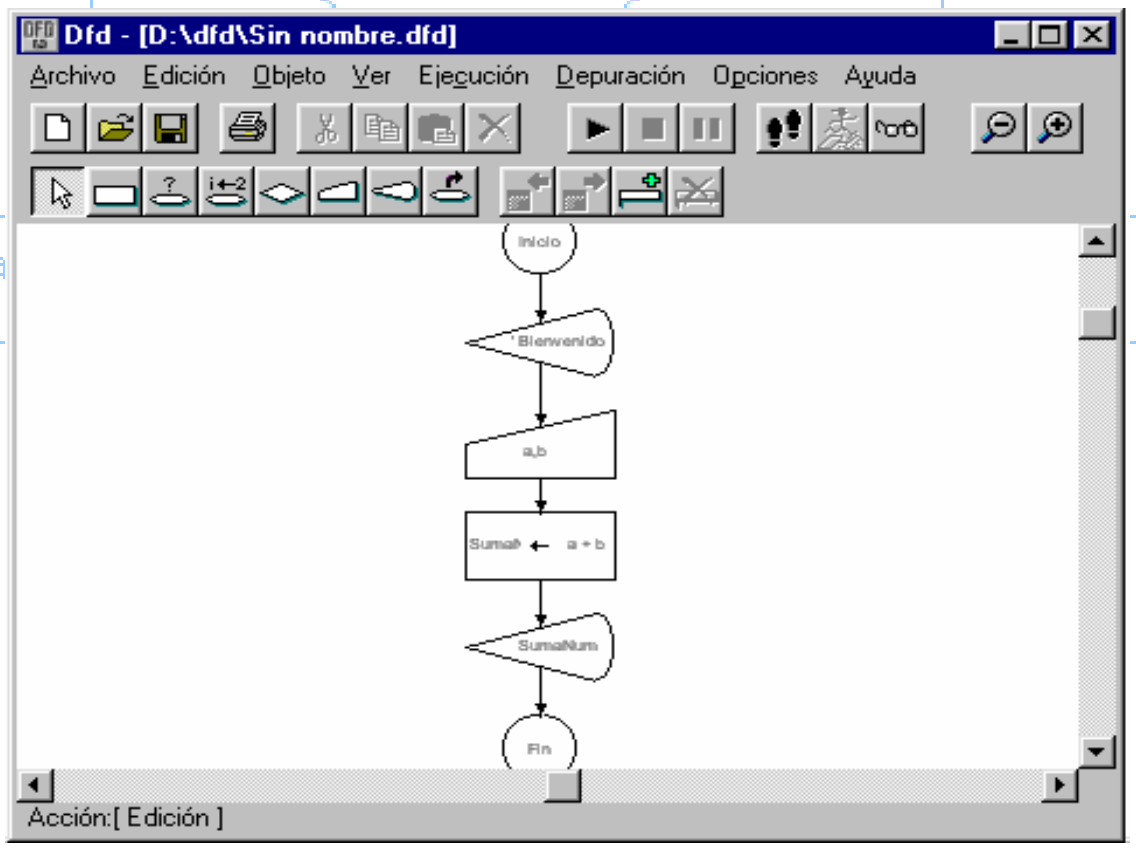
Paso 5.- Reporte. Después del símbolo de Proceso  inserte el símbolo de Reporte 

o el símbolo de salida  donde se define la variable de salida SumaNum (si desea puede imprimir también los números de entrada.)

Hacer doble clic luego se muestra el formulario, para editar la variable respectiva. Se ilustra en la siguiente grafica.



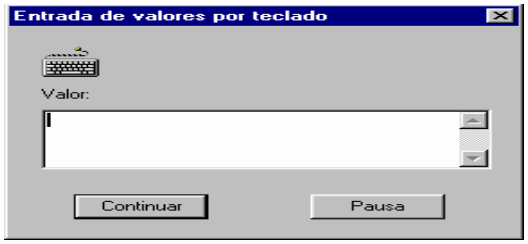
Así finaliza la edición del diagrama de flujo y quedando lista para su ejecución.



EJECUCION DEL DIAGRAMA DE FLUJO DE DATOS

Es la parte final donde se vera resultados de la suma de 2 numero, NO en el diagrama sino en diferentes formularios. Veamos:

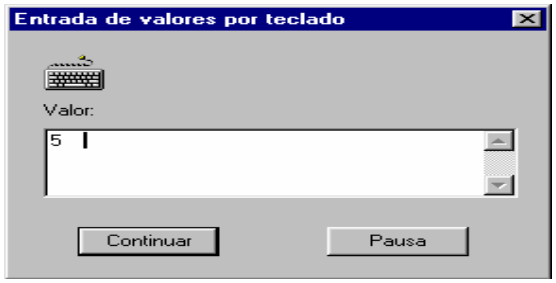
Paso E1.- Usar el Símbolo de Ejecución (compilación). El programa envía primero el mensaje, tal como se ilustra en la siguiente figura.



Paso E2.- Ingreso de datos.- Presione Continuar y luego observara el formulario para entrada (ingreso) de datos.

Por cada dato que Ud. Ingrese presione **Continuar,** por ejemplo

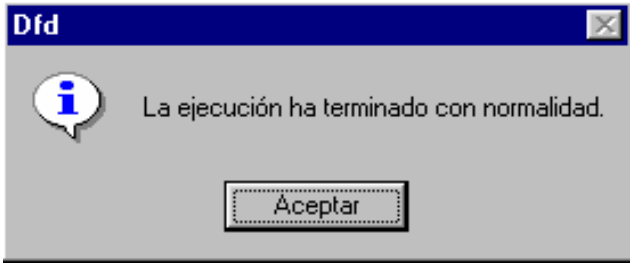
En el primer formulario ingrese 10 y luego continuar.



Luego aparece el segundo formulario ,ingrese 5 , tal como se ilustra en el formulario adjunto.

Finalmente presione Continuar y obtendrá su resultado, tal como se ilustra en la siguiente grafica.

Finalmente el sistema informa que el programa finalizo correctamente. En la siguiente figura se ilustra la confirmación

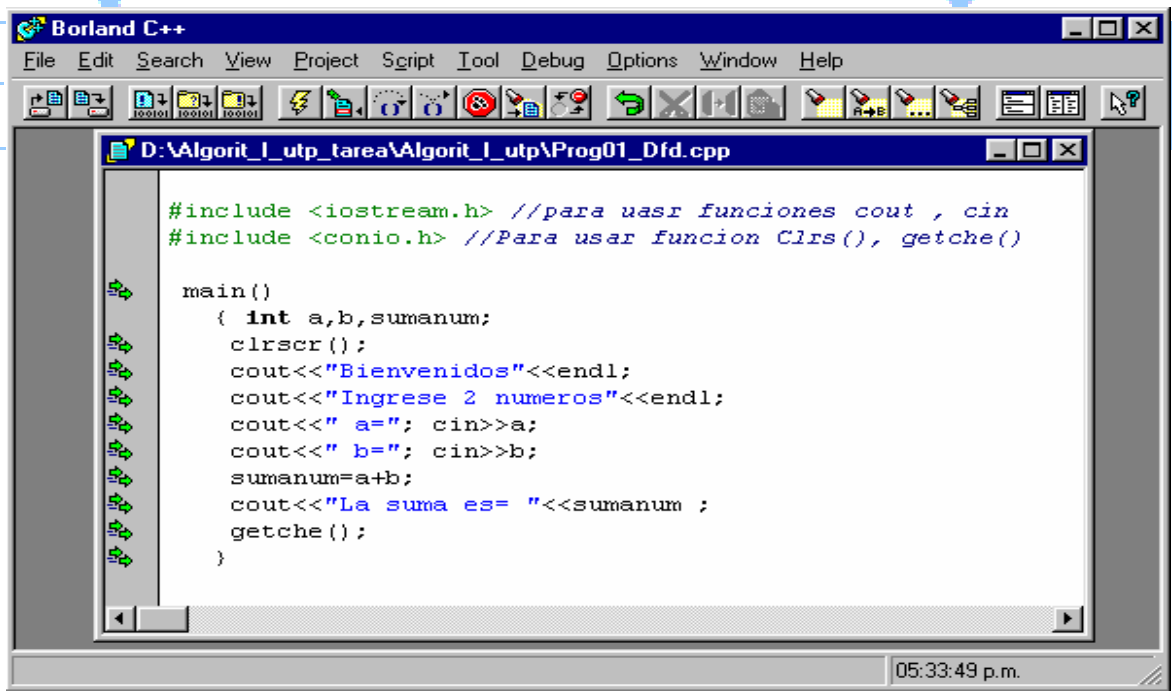


Sr. Lector, se ha ilustrado los pasos correctamente usando un ejemplo básico, pues el objetivo inicial es manejar con destreza el software y aplicarlo en otros programas de mayor complejidad como se ilustraran en otros ehjemplos, pero los pasos a ilustrarle serán mínimos.

VERIFICACION.- La verificación puede hacerlo en forma manual, pero para fines de expresar la Lógica del programa anterior en Sentencias de un Lenguaje de Programación, en este momento ilustrare usando Borland C++ 5.0.

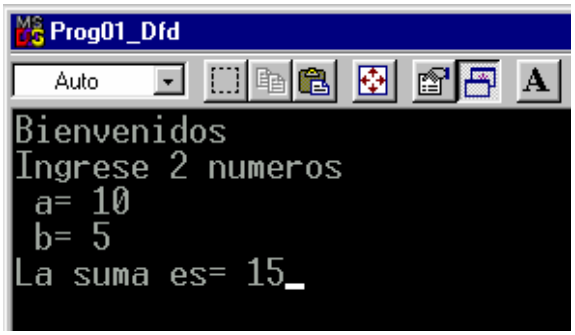
1.- Usando Lenguaje de Programación Borland C++ 5.0

En la siguiente figura, se ilustra el Programa Fuente



A continuación use el botón  para ejecutar el programa, los resultados se observa a continuación.

Resultados que coinciden el ejecutar el diagrama de Flujo.

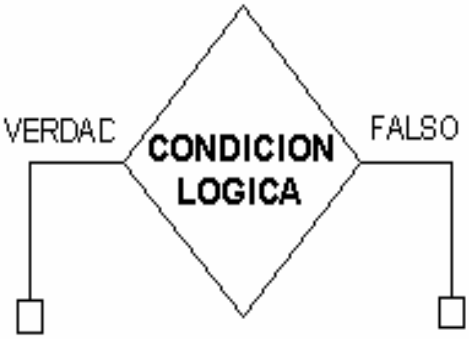


2.- Estructuras Condicionales

Estructuras Selectivas: Si … Entonces … Sino

Su uso permite evaluar una Condición para luego ejecutar una sola tarea.

Sintaxis




```
Si < Condición_Lógica > entonces
        <Instrucción1>
Sino
        <Instrucción2 >
```

```
Si <Condición_Lógica> entonces
   Inicio
      <Instrucciones1>
   Fin
Else
   Inicio
      <Instrucciones2>
   Fin
```

Problema # 1.- Diseñar un Diagrama de Flujo que permita leer 2 números enteros positivos **m** y **n,** luego:

a).- Calcula el producto de m por n si el numero m es mayor que el numero n

b).- Calcula la raíz cuadrada si el numero m es menor que n

Solución

Primero .- Se ilustra la solución mediante un programa **Pseudocodigo (imitación de instrucciones maquina):**

Inicio

Imprimir("Lectura de 2 números m y n ")

Imprimir("ingrese numero m ="), leer(m)

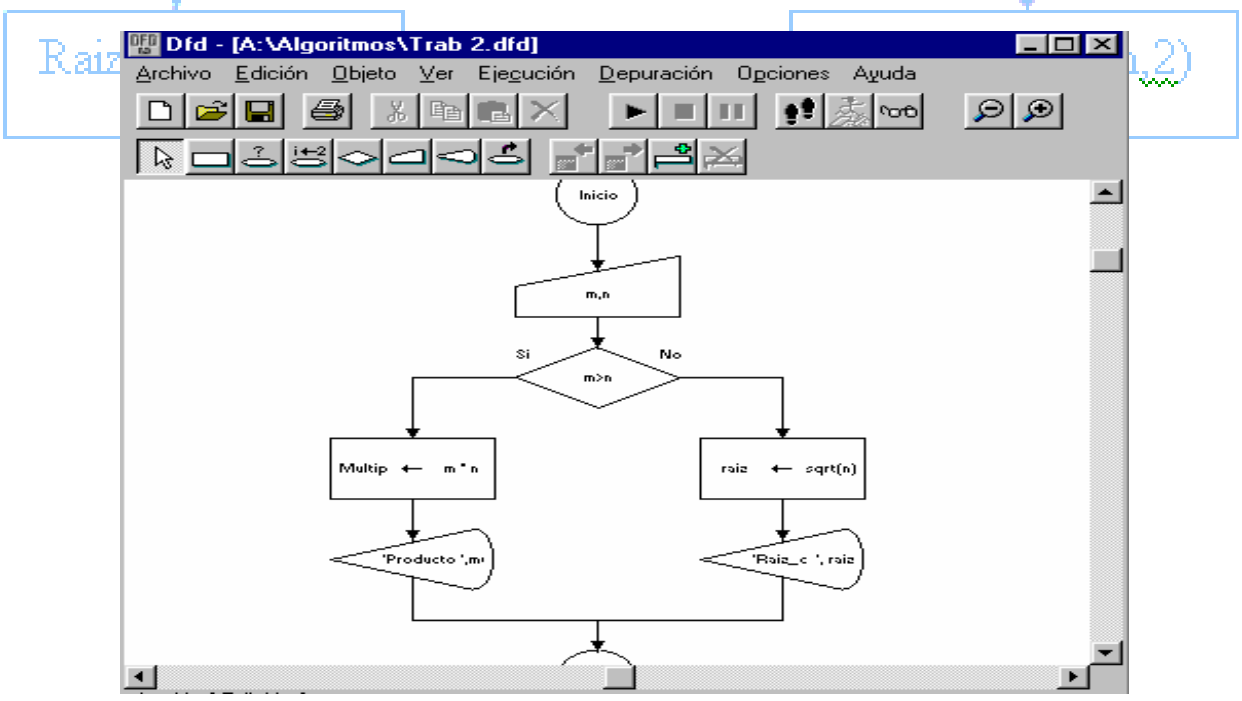
Imprimir("ingrese numero n ="), leer(n)

si (m> n) entonces
inicio
mult <------m*n
imprimir("la multiplicación es = ",mult)
fin
sino
inicio
raiz<------ sqrt(n)
imprimir("La raíz cuadrada es = ",raiz:10:4)
fin

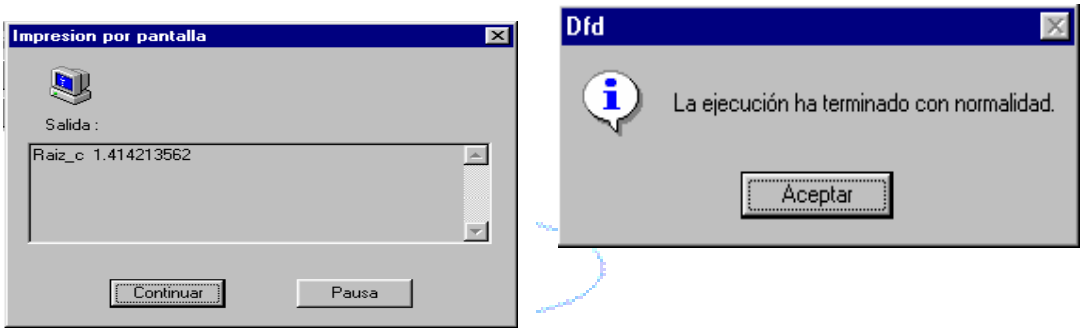
Fin

Análisis: para m = 1, n = 2, el programa solo ejecuta el bloque correspondiente a la condición FALSO, pues m>n (1>2). Imprimiendo finalmente: **imprimir("La raíz cuadrada es = ", raiz:10:4)**

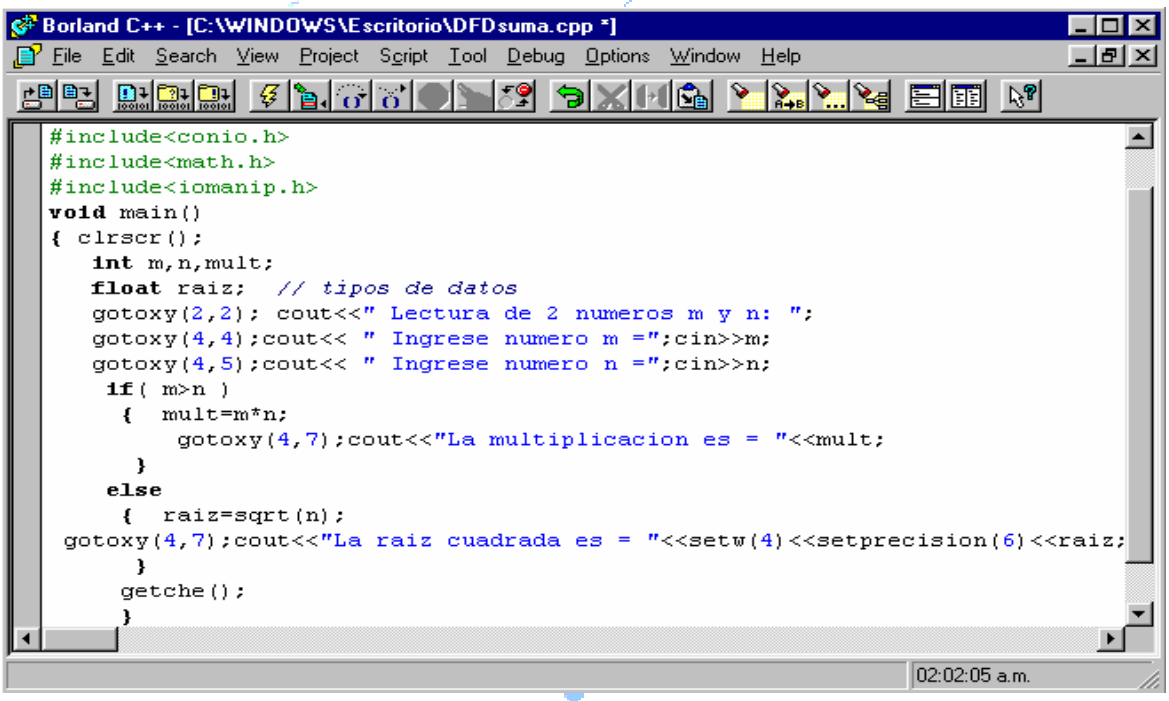
Segundo.- Mediante El Diagrama de flujo (DFD).



Ejecución: como el resultado de evaluar la condición es **Falso,** entonces se calcula la raíz cuadrada del número n y envía el mensaje de conformidad

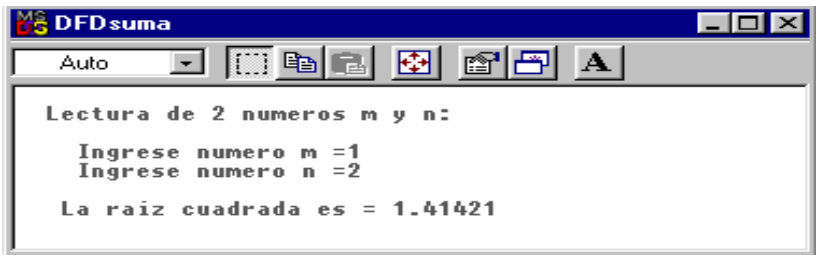


Tercero.- Usando Lenguaje de Programación Borland C++ 5.0, en la siguiente grafica se ilustra el **programa fuente(PF).**



La ejecución de programa, se ilustra en la siguiente grafica

Problema # 2.- Diseñar un Diagrama de Flujo que permita a un alumno



ingresar su código =001 y su clave = 1010. Luego si los datos son correctamente ingresados el programa permite ingresar 3 practicas calificadas pc1,pc3 y pc3, luego calcula el promedio y muestra el reporte respectivo. Si los datos del alumno son incorrectos, debe emitir un mensaje "Sr. Alumno, Errores en datos"

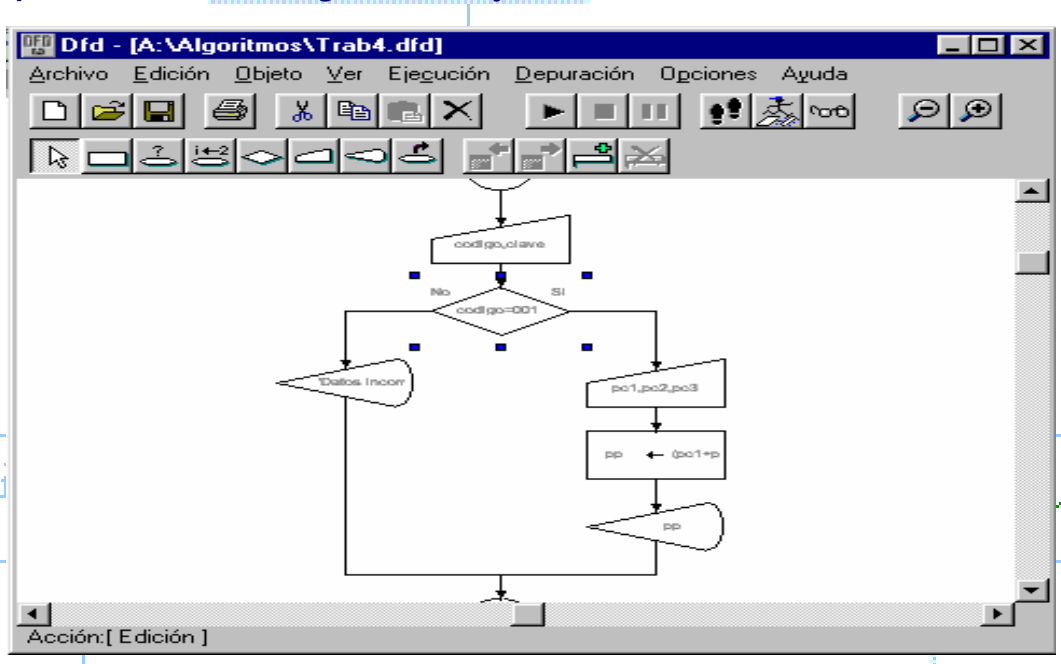
Solución.- La condición a validar es

Si (codigo=001) y (clave = 1010) entonces ' Leer 3 practicas y calcular su Promedio'

Sino

Imprimir 'debe emitir un mensaje "Sr. Alumno, Errores en datos"

Implementación del Diagrama de flujo'.



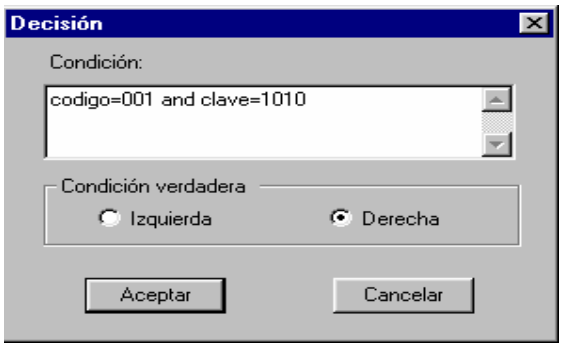
Ejecución:

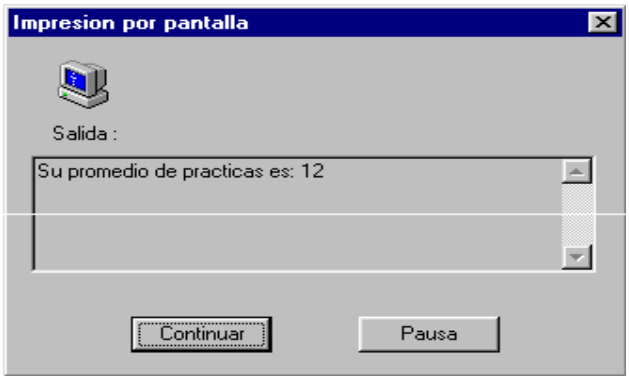
1.- Ingresar el código y la clave correctamente

2- Ingresar las 3 prácticas calificadas.

Pc1=12, Pc2=10 y Pc3=14.

El promedio de practicas, se ilustra en la siguiente grafica, así como la conformidad





Estructuras Condicional con Anidamiento

Sintaxis

```
Si  <cond_logica1> entonces
        Inicio
                <accion1 >
        fin
sino
        Si  <cond_logica2> entonces
                Inicio
                   <accion2 >
                fin
sino <accion3>
```

Problema # 3. Diseñar un diagrama de flujo que permita ingresar 3 números enteros a, b y c. luego el programa averiguar si se cumple una de las tres relaciones y relación que se cumple ejecutar su tarea respectiva. Veamos.

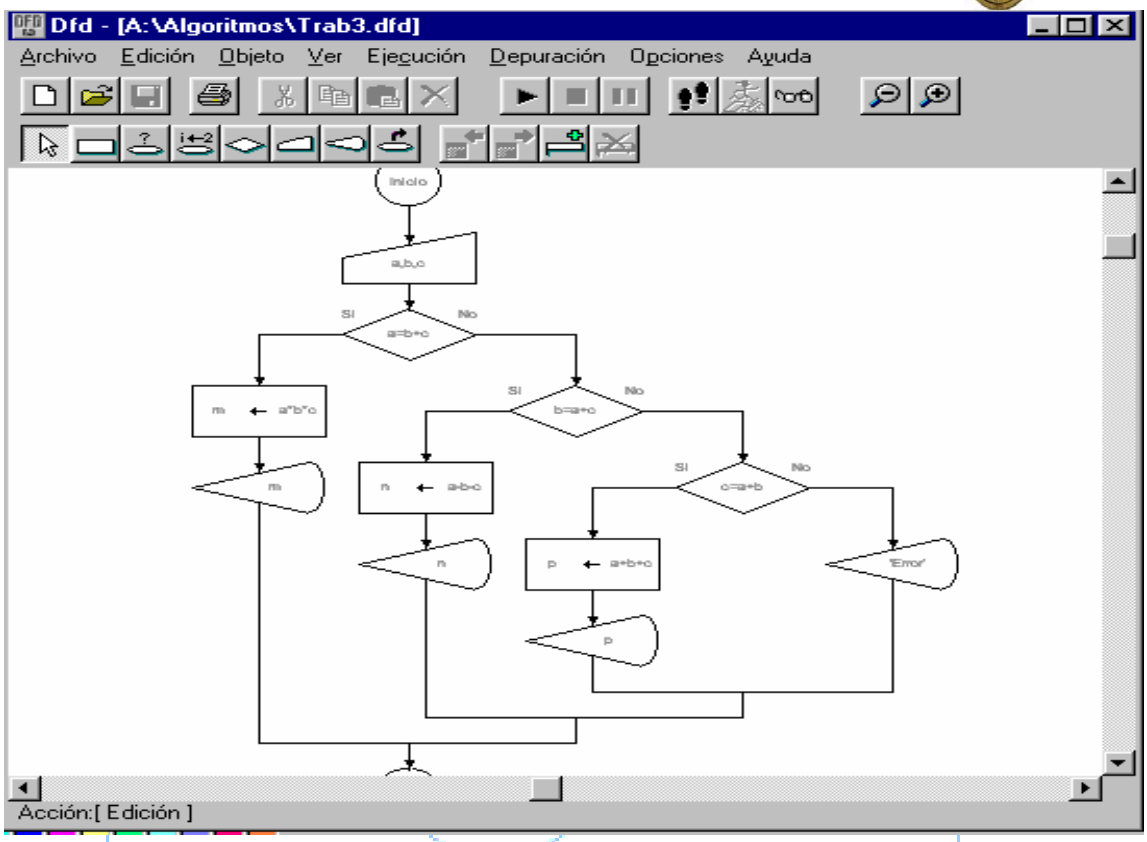
Relación 1.- Si a = b + c, entones calcular el producto de los 3 números

Relación 2.- Si b = a + c entones calcular la resta de los 3 números

Relación 3.- Si c = a + b entones calcular la suma de los 3 números

Si ninguna relación se cumple, emitir un mensaje: ‘ Sr. No existe relación ’

Solución. En la presente grafica, se ilustra el diagrama de flujo, que realiza la tarea respectiva.



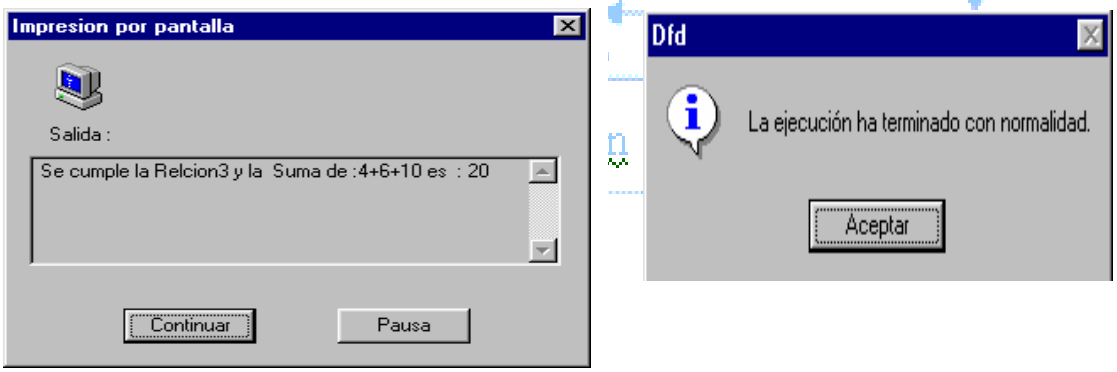
Ejecución.-

Lectura de datos: considere la lectura de los números : a = 6, b = 4 y c = 10.

Proceso: La relación que se cumple es la relación 3, entones el resultado será:

m = a + b + c = 20.

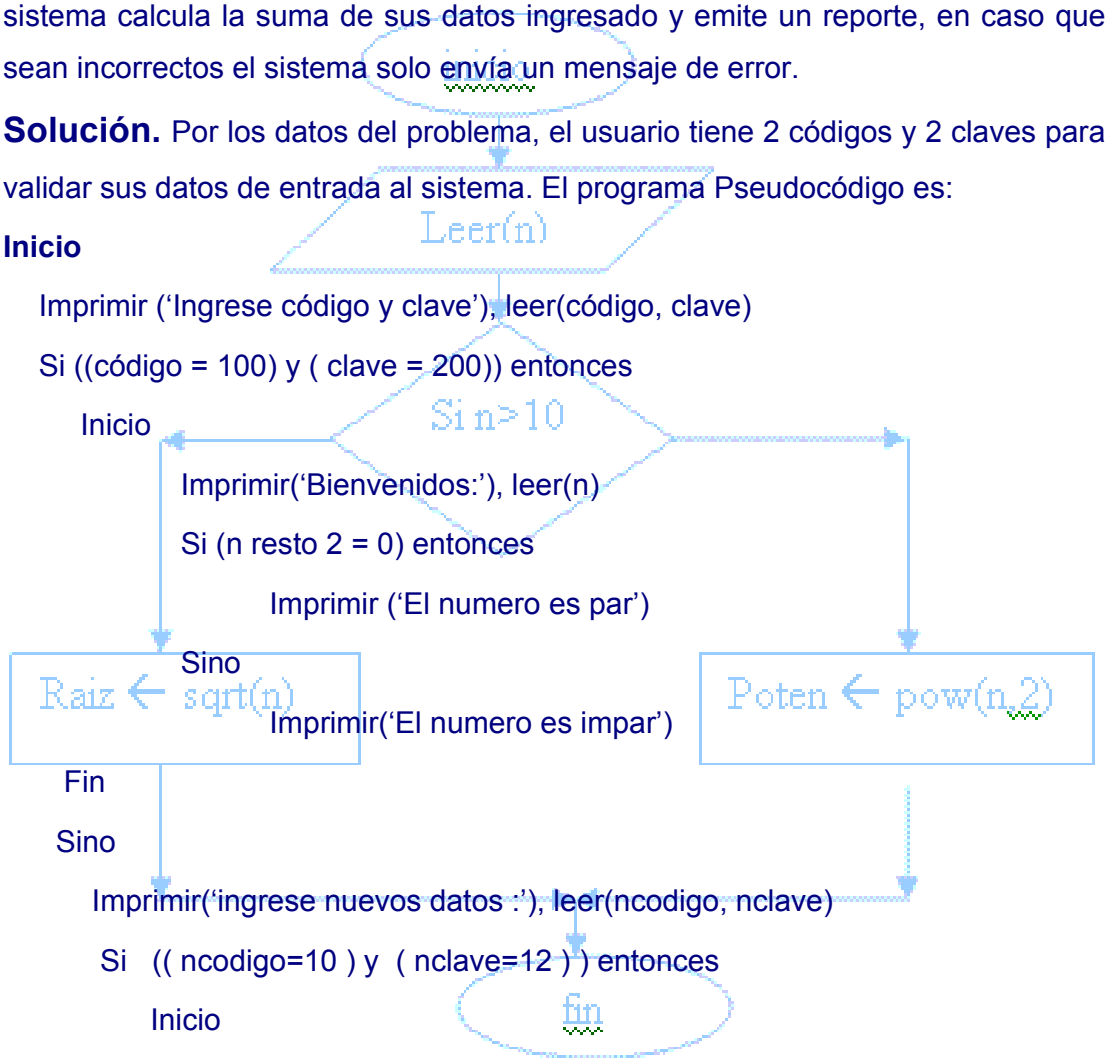
Salida : Los resultados se ilustran en la siguiente grafica



Problema # 4. Diseñar un diagrama de flujo que permita ingresar a un usuario código = 100 y clave =200. Silos datos de entrada son correctos, el sistema solicita al usuario que ingrese un numero n y verifica si este numero es par o impar y luego envía el reporte según se el caso.

Si el usuario no recuerda sus datos en la primera entrada, decide ingresar sus nuevos datos mediante ncodigo=10 y nclave=12. Si estos son correctos el sistema calcula la suma de sus datos ingresado y emite un reporte, en caso que sean incorrectos el sistema solo envía un mensaje de error.

Solución. Por los datos del problema, el usuario tiene 2 códigos y 2 claves para validar sus datos de entrada al sistema. El programa Pseudocódigo es:



Inicio

Imprimir (‘Ingrese código y clave’), leer(código, clave)

Si ((código = 100) y (clave = 200)) entonces

Inicio

Imprimir(‘Bienvenidos:’), leer(n)

Si (n resto 2 = 0) entonces

Imprimir (‘El numero es par’)

Sino

Imprimir(‘El numero es impar’)

Fin

Sino

Imprimir(‘ingrese nuevos datos :’), leer(ncodigo, nclave)

Si ((ncodigo=10) y (nclave=12)) entonces

Inicio

Suma_codigo=ncodigo+ nclave

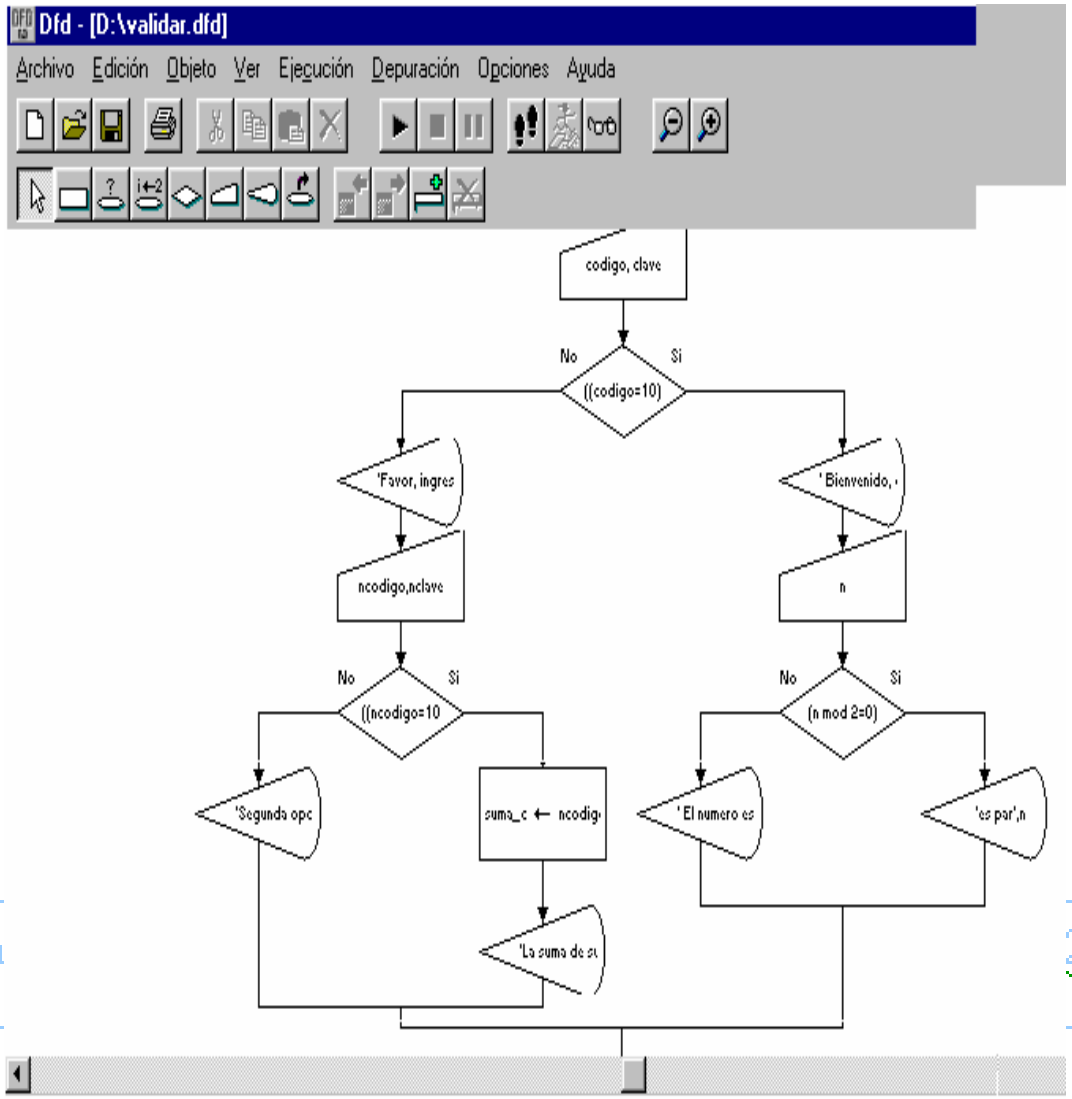
Imprimir(Suma_codigo)

Fin

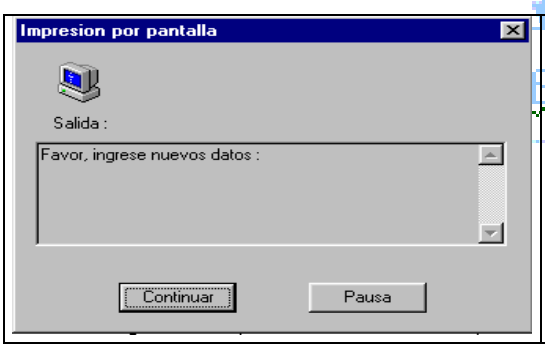
Sino Imprimir(‘Segunda opción, error :’)

Fin

Diseño del Diagrama de Flujo

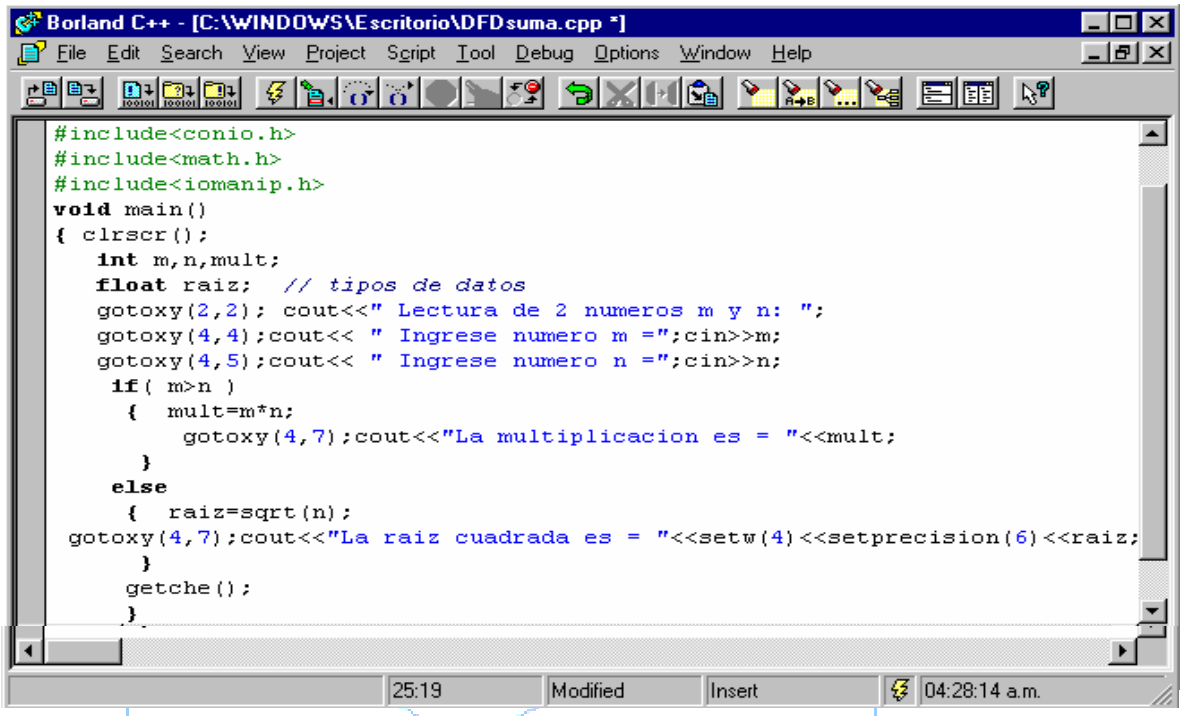


Ejecución.- Si sus primeros datos contienen errores, entonces le solicita los segundos datos y a estos los acumula. Ver la siguiente grafica.

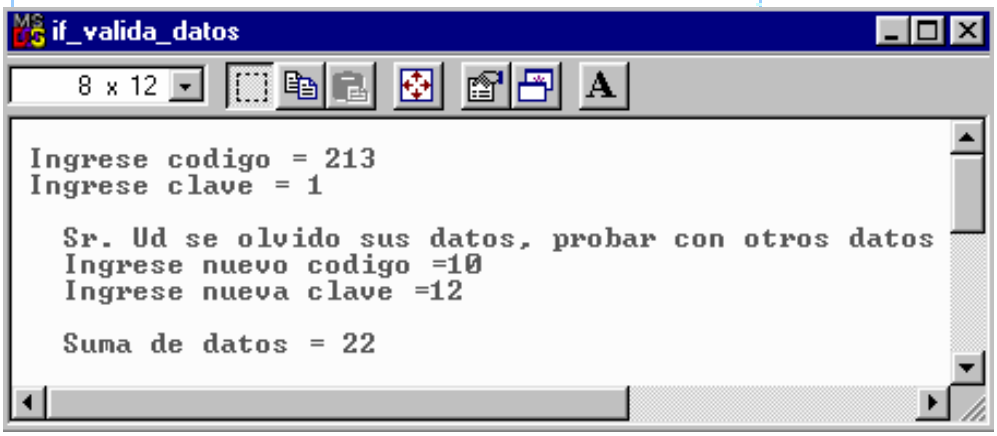


Si Ud. ingresa datos de validación diferentes a 100 y 200, obtendrá el mensaje de ingresar segundo tipos de datos y con estos se mostrara la suma acumulada de 10 +12.

Usando Lenguaje de Programación Borland C++ 5.0, en la siguiente grafica se ilustra el **programa fuente(PF).**



Después de ejecutar el programa fuente, se obtiene los resultados mostrados en la siguiente grafica, resultados después de haber cometido errores en la primera entrada de datos de validación



:

3.-Estructuras de Control Repetitivas.- Permiten ejecutar sentencias hasta satisfacer una condición Lógica.

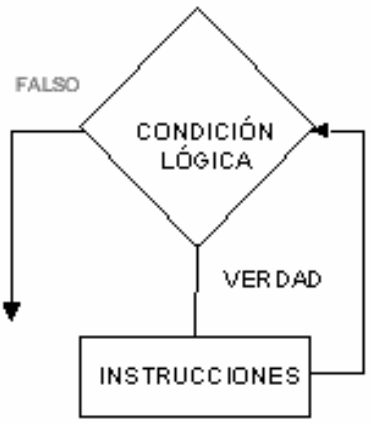
3.1 Repetitiva con Entrada Controlada: Mientras

Realiza 0,1 2,.., o mas tareas.

Es 0 tareas cuando la condición de evolución da como resultado FALSE

Es 1,2...,n tareas cuando la condición de evaluación de cómo resultado TRUE.

Sintaxis




```
Mientras <CondiciónLógica> hacer
        Instrucción;
```

```
Mientras <CondiciónLógica> Hacer
   Inicio
        <Instrucciones>
   Fin;
```

Problema # 1. Diseñar un diagrama de flujo que permita calcular la suma acumulada de la serie: 1+2+3+4+5.

Solución.

a).- Se define un **contador (cont)** para que incremente elementos a sumar y un **acumulador (acum)** que vaya acumulando los valores según como avance el contador.

b).- La condición de terminación esta controlado por el contador, el cual avanza hasta que se menor o igual a 5.

c).- Se debe inicializar el contador en 1 y el acumulador en 0

Solucion

Mediante Programa Pseudocódigo.

Inicio

Imprimir(‘ Bienvenidos :’)

Cont=1

acum.=0

Mientras (cont<=5) hacer

Inicio

Acum = acum.+ cont

Cont = Con t+ 1

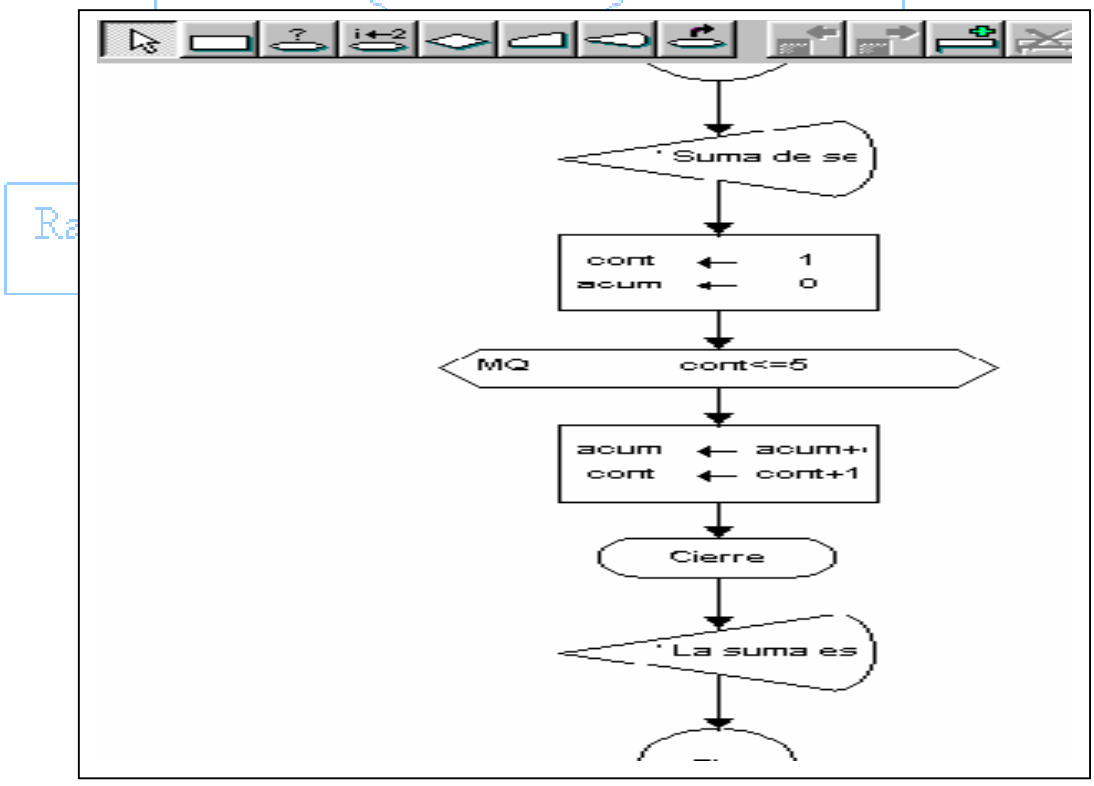
Fin

Imprimir (‘La suma acumulada hasta 5 es : ’,Acum)

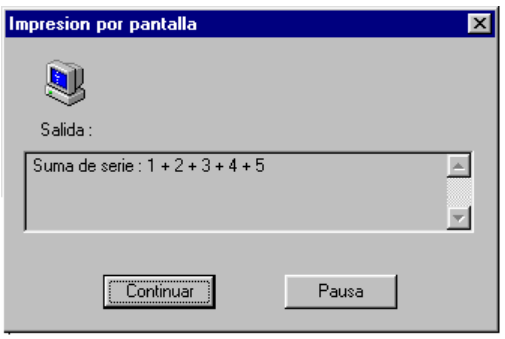
Fin.

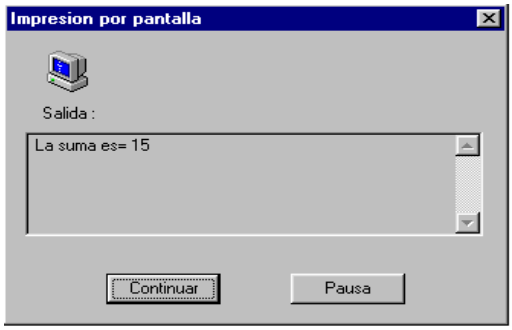
Ahora se diseña el **Diagrama de Flujo**

Ejecución.-



Los resultados se ilustran en la siguiente grafica y su confirmación respectiva.





Problema # 2. Diseñar un diagrama de flujo que permita ingresar al usuario un primer elemento de la serie y luego un segundo elemento de tal manea que ambos elementos definen un grupo de elementos de la serie(rango) y luego calcular la suma acumulada de la serie: 1+2+3+4+5+ 6 +……+ n..

Solución.

a).- Se define un **Contador (Cont_I)** para que defina lado izquierdo de la serie y **Lado_d,** lado derecho como segundo elemento de la serie. Ambos elementos definen un intervalo, bajo el cual se calcula la suma. Asimismo se define el **acumulador (Acum)** que va acumulando los valores según como avance el contador.

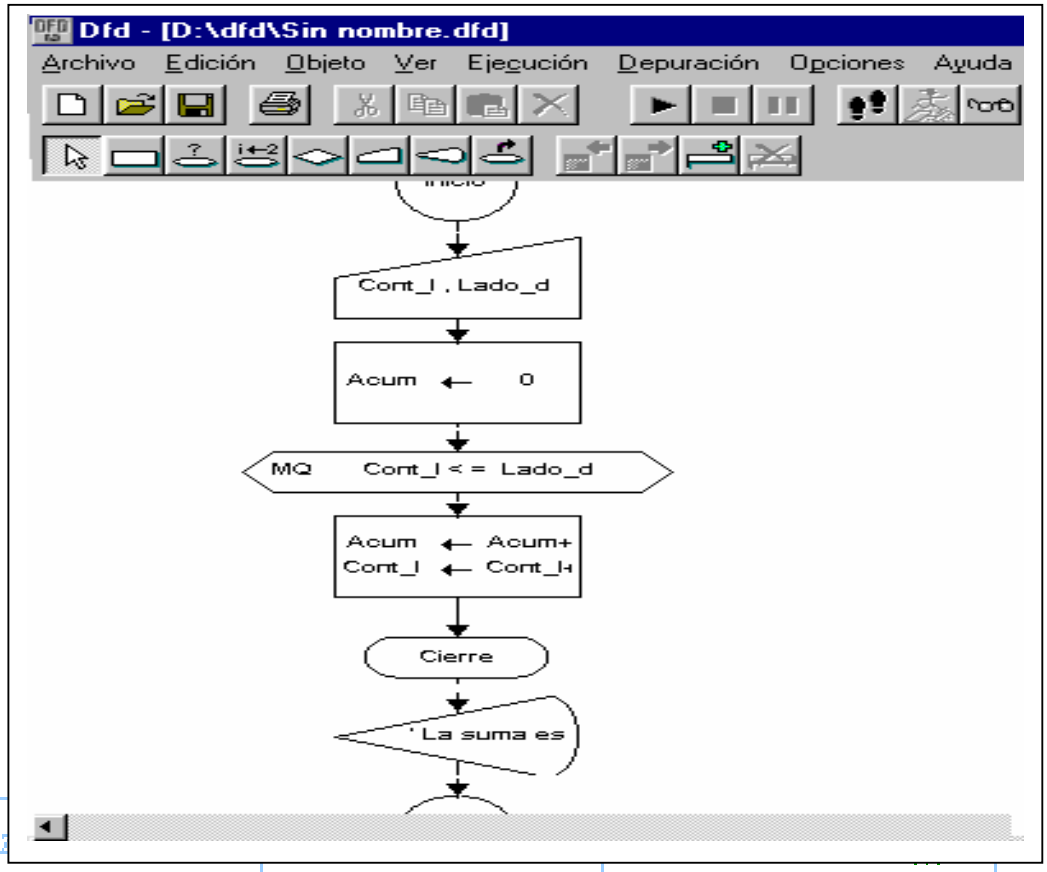
b).- La condición de terminación esta controlado por: **Cont_I <= Lado_d** , ambos deben leerse desde el teclado

c).- Se debe inicializar el acumulador en 0

Mediante un Programa Pseudo código.

```
Inicio
   Imprimir(' Ingrese lado izquierdo y derecho de la serie:'), leer(Cont_I, Lado_d)
  acum.=0
  Mientras (Cont_I<=Lado_d) hacer
      Inicio
              Acum = acum.+ Cont_I
              Cont_I = Con t_I+ 1
      Fin
  Imprimir ('La suma  es =: ',Acum)
Fin
```

Ahora se diseña el **Diagrama de Flujo**

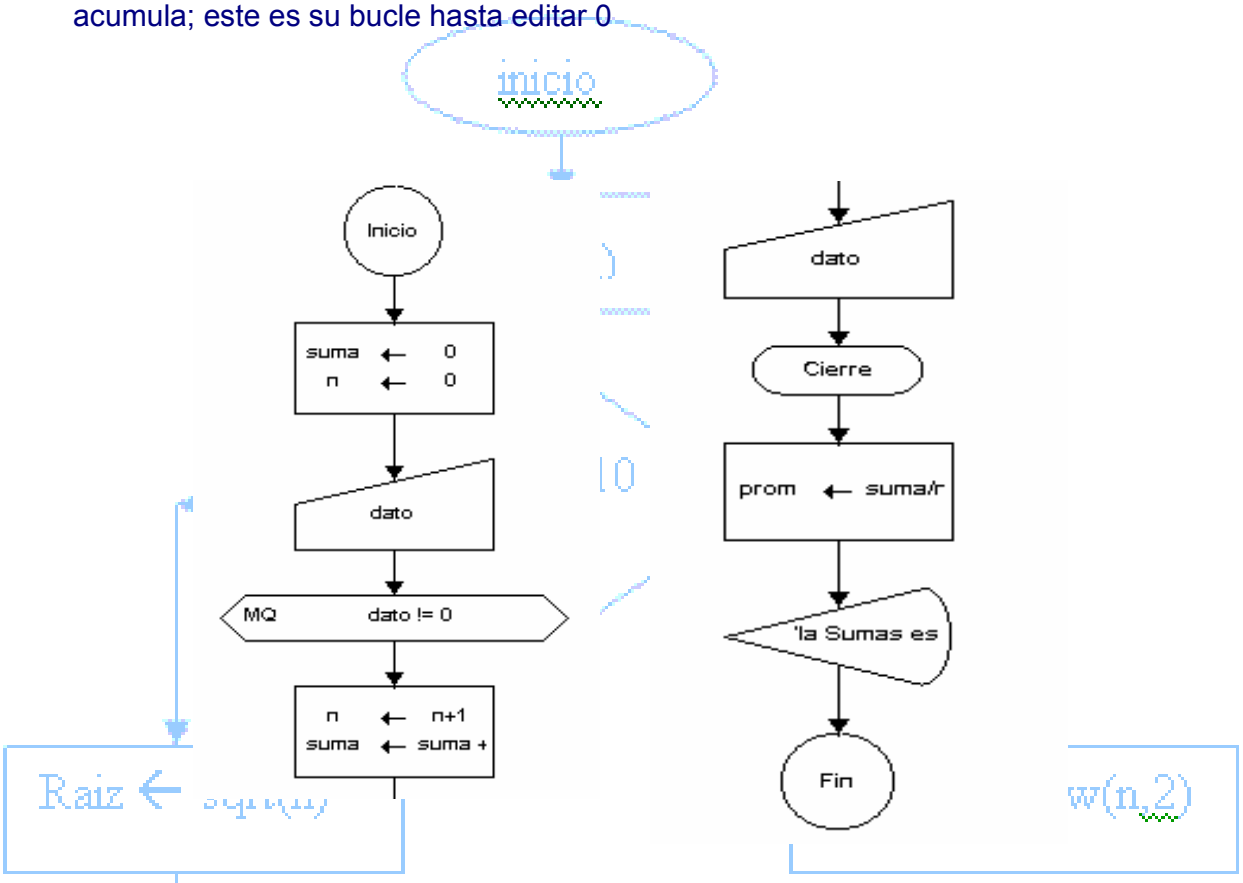


Ejecución.- usando Cont_I=3 y Lado_d=5 , el resultado debe ser: 3+4+5=12.

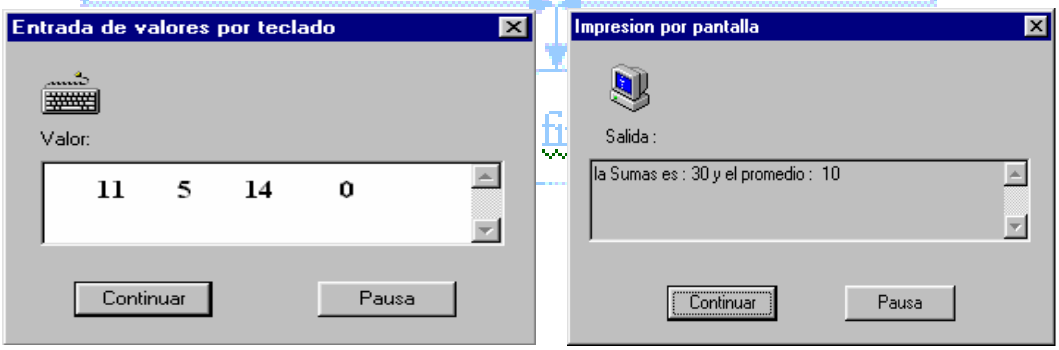
Los resultados se ilustran en las siguiente graficas y su confirmación.

Problema # 3. Diseñar un diagrama de flujo que permita ingresar datos tipo entero y luego calcular la sumatoria y el promedio del conjunto de datos. Finaliza el ingreso de notas si edita 0.

Solución. El programa solicita un dato, verifica si es diferente de cero, entonces acumula; este es su bucle hasta editar 0

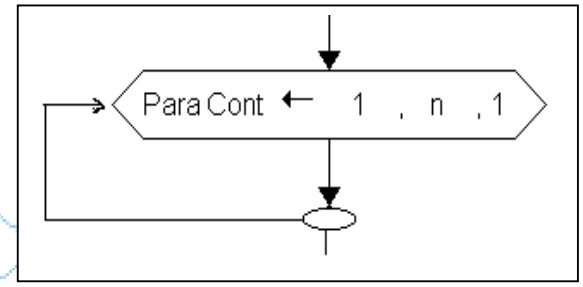


Ejecución .- Edite 11 5 14 , los resultados se ilustran a continuación,



3.2 Estructura de Control con número de instrucciones conocidas

Se conoce el número de instrucciones (bucles) a realizar



Desde Vc <-- Vi hasta Vf Hacer

Inicio

<Instrucciones>

Fin

Observación:

1.- Ascendente : Se debe cumplir que Vi <= Vf

2.-Descendente: Se debe cumplir que Vf>=Vi

Problema # 1. Diseñar un diagrama de flujo que permita leer el numero de términos de la serie 1+2+3+4+5+……+n y luego calcular la suma acumulada.

Solución.

a).- Se define un **contador (cont)** como variable de control y **n** el numero de términos a ingresar. Para sumar se define la variable **Acum** que va acumulando los valores según como avance el contador.

b).- La condición de terminación esta controlado por el contador, el cual avanza hasta que se menor o igual a **n,** dato ingresado por el usuario.

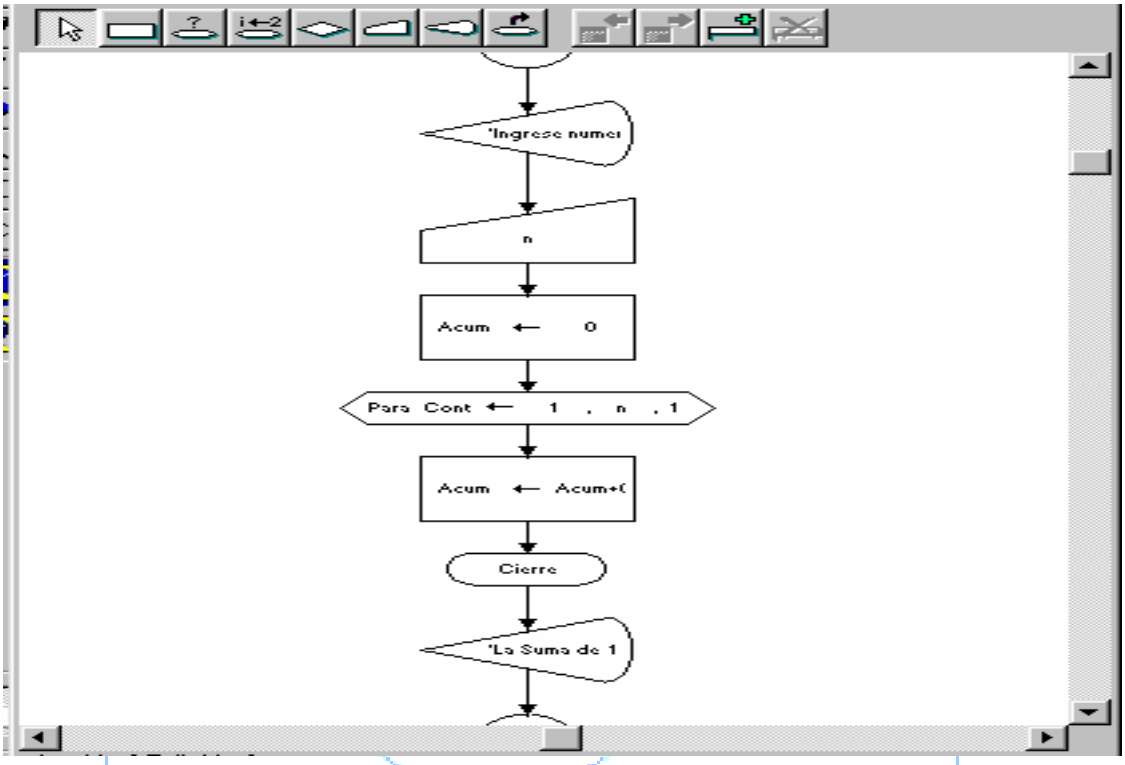
c).- Se debe inicializar el Acumulador en cero.

Mediante Programa Pseudocódigo.

```
**********************************************************************************************
Inicio
   Imprimir(' Ingrese Numero de Terminos')
  Acum.=0

  Para cont←1 Hasta n hacer
      Inicio
              Acum = acum.+ cont
      Fin
  Imprimir ('La suma acumulada hasta  : ',n , ' es =', Acum)
Fin.
```

Ahora, mediante **Diagrama de Flujo**

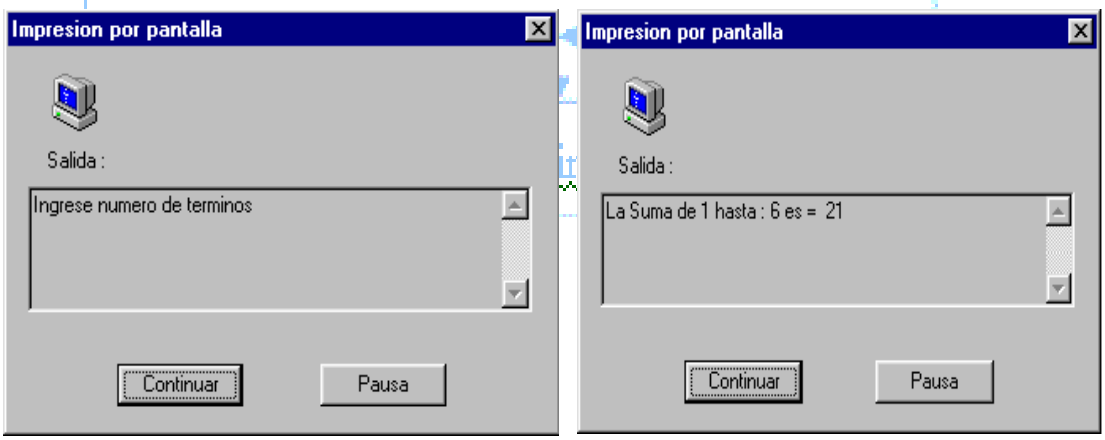


Ejecución.- Ingresando numero de elementos n=6
Acum: 1+2+3+4+5+6 = 21

En la siguiente grafica, se ilustra los procedimientos de ejecución.

Lectura y Proceso:

Al ejecutar, el programa le solicita que ingrese el número de elementos de la serie a sumar, para nuestro caso ingrese **6.**

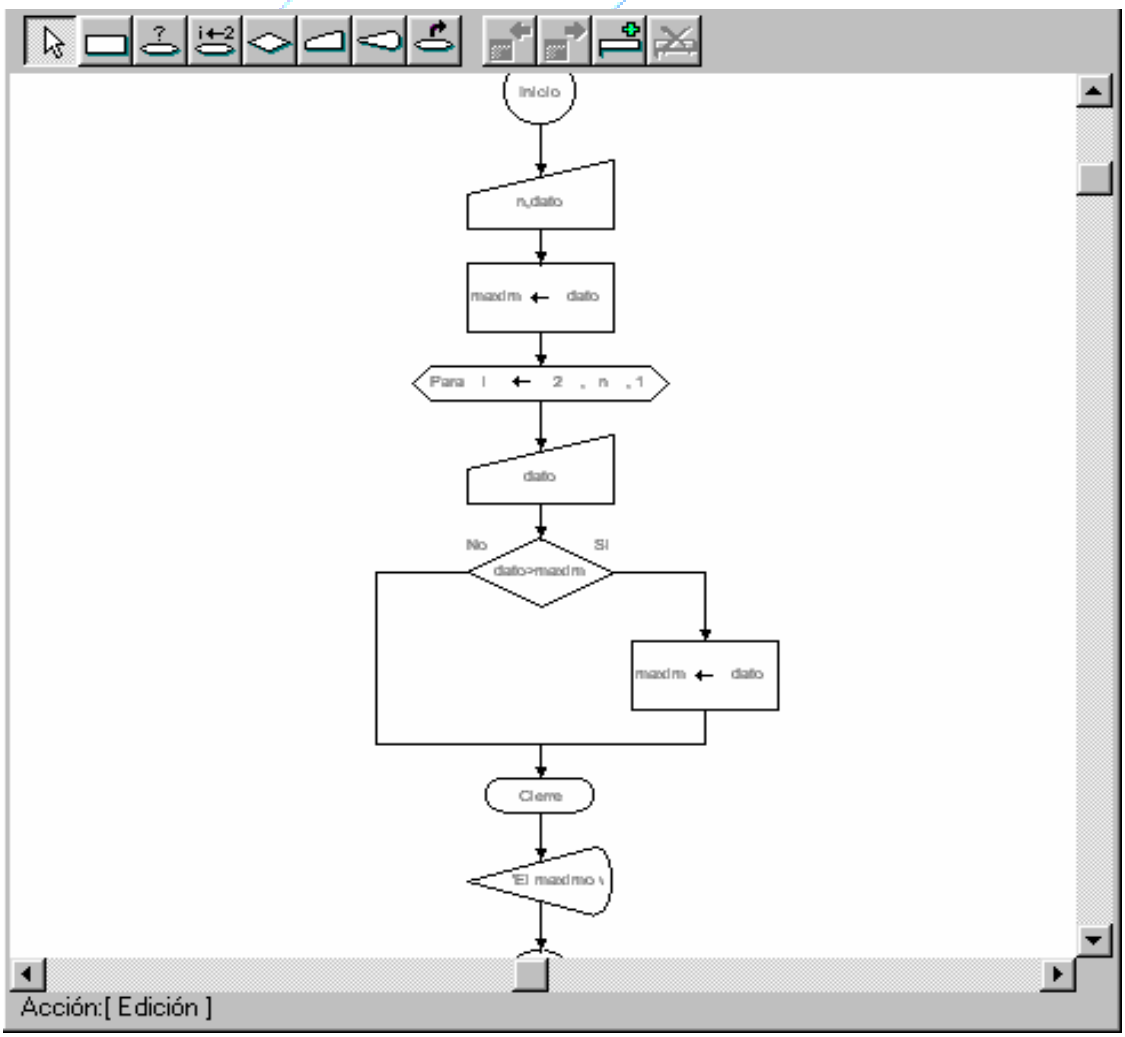


Problema # 2. Diseñar un diagrama de flujo que permita leer n datos de tipo entero y luego genere un reporte que muestre el número mayor.

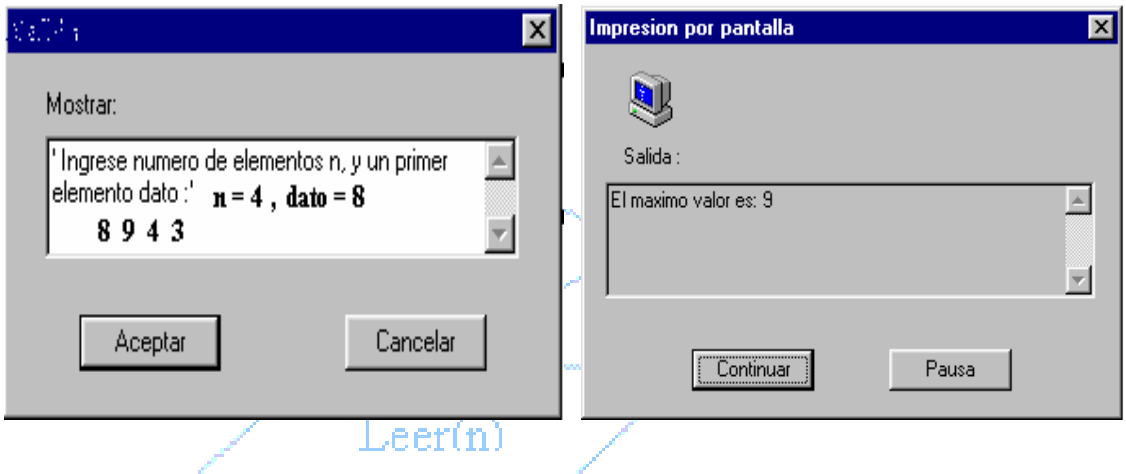
Solución. Se definen las siguientes variables: n : para leer el número de elementos. Dato : para lectura de los elementos. Máximo : para almacenar el numero mayor.

Se usa la estructura de control repetitiva Para……, que permite leer desde el segundo elementos hasta el total y luego se usa la estructura condicional si….., para comparar dos datos y si es verdadera la respuesta asigne o cargue el datos la variable máximo. Finalmente se imprime la variable máximo.

Ver diagrama de flujo siguiente:



Ejecución:



Problema # 3. Diseñar un diagrama de flujo que permita conocer el numero total de puntos que se encuentran en el interior de la elipse:

X*x/16 + y*y / 9=1

Solución.

Se definen las siguientes variables:

Punto : para contar el total de puntos (x,y)

Radio: para verificar la condición : radio<1 entonces existe punto interior a la elipse.

Se definen 2 bucles:

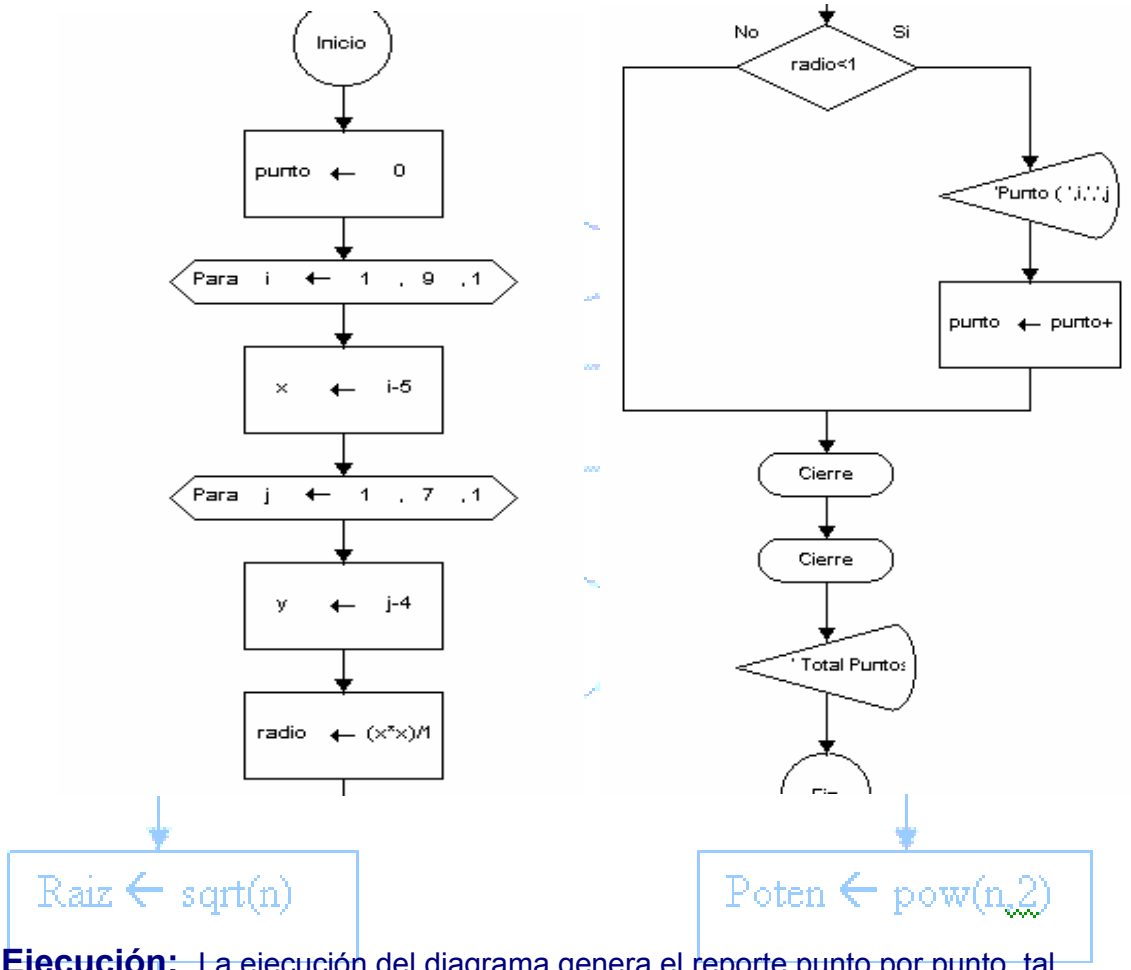
Un bucle externo y un Bucle Interno

El bucle externo:

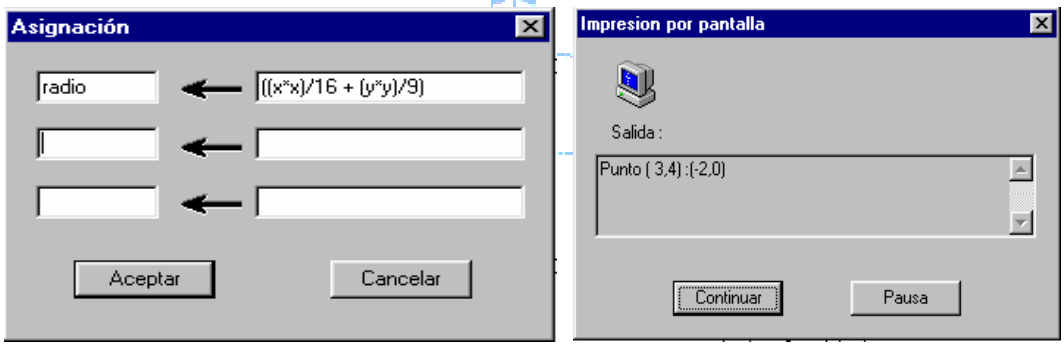
esta definido para cada valor de las abcisas (puntos en el eje x)

El bucle Interno esta definido para cada valor de las ordenadas (puntos en el eje y).

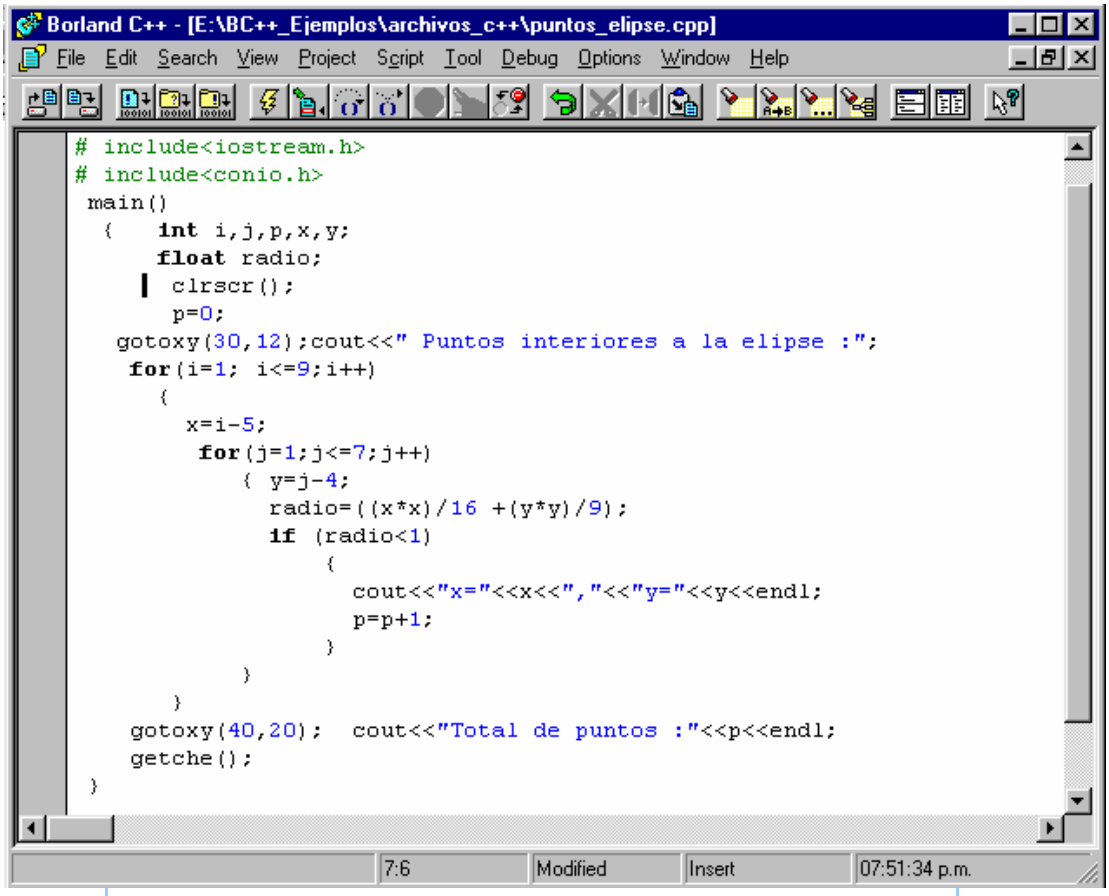
A continuación se ilustra el diagrama de flujo.:



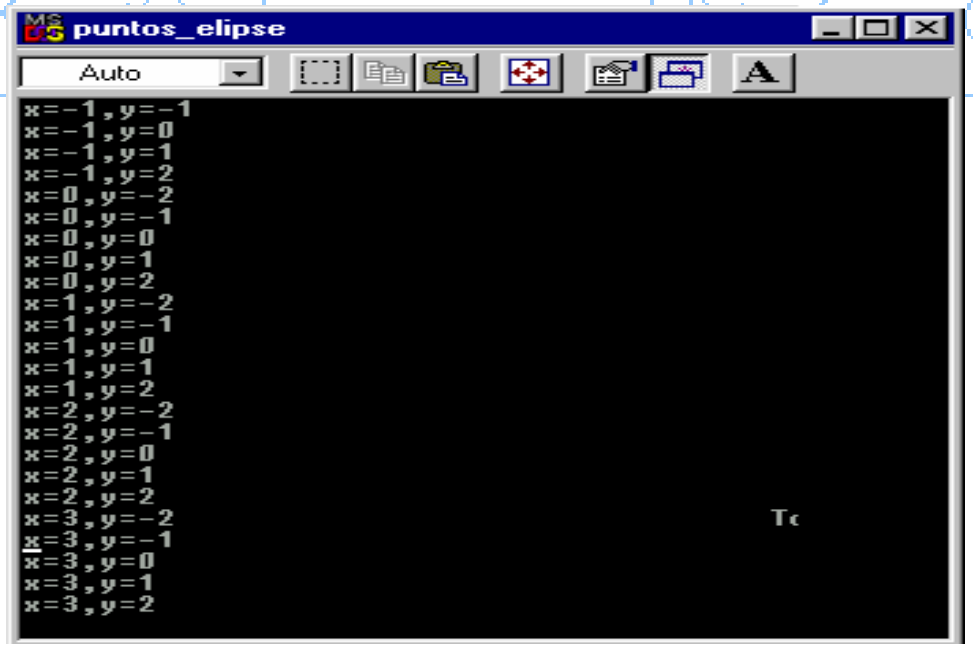
Ejecución: La ejecución del diagrama genera el reporte punto por punto, tal cmo se ilustra en la siguiente grafica.



Usando Borland C ++ 5.0

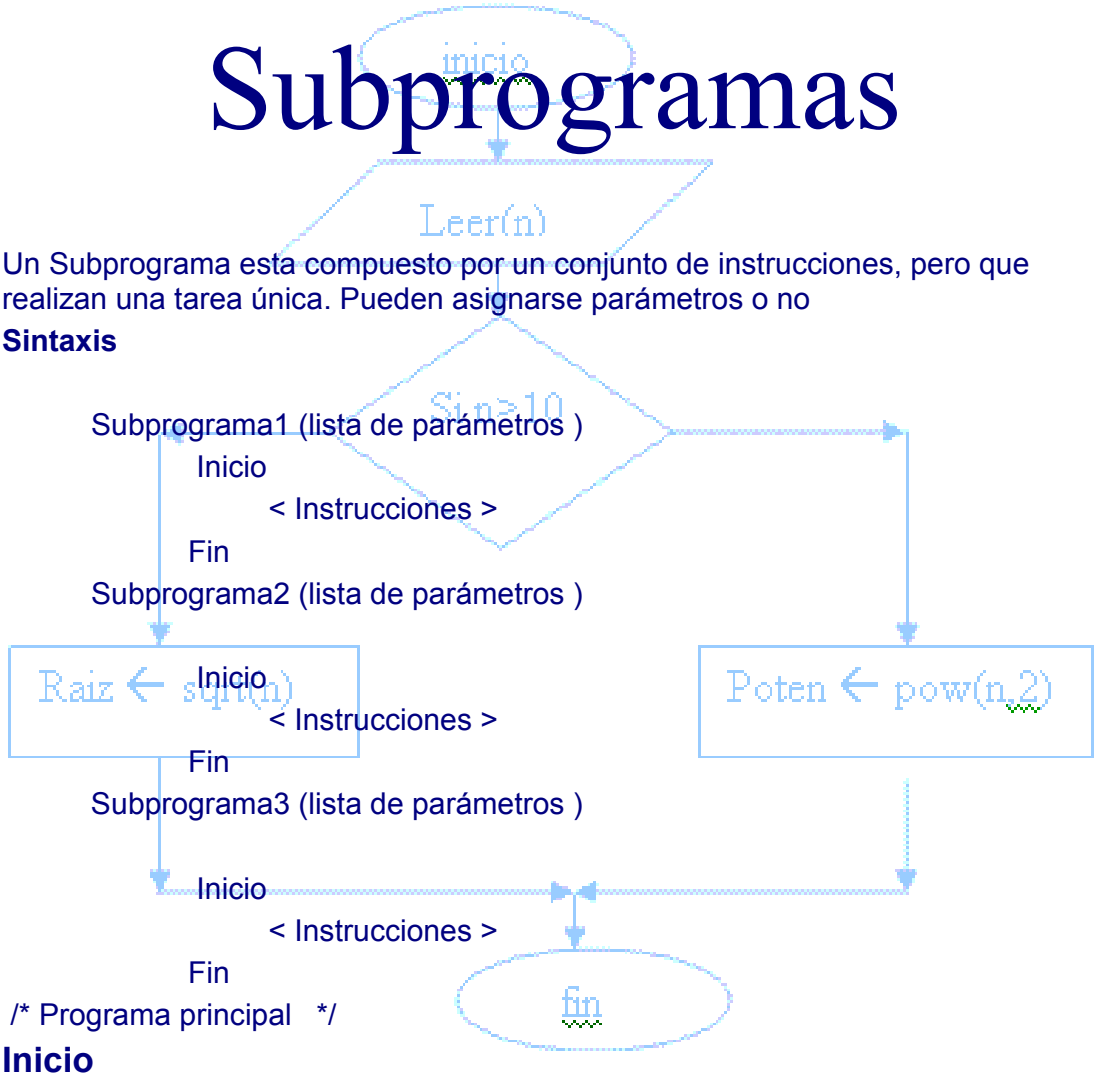


Ejecución:



Diseño Modular

Subprogramas



Un Subprograma esta compuesto por un conjunto de instrucciones, pero que realizan una tarea única. Pueden asignarse parámetros o no

Sintaxis

Subprograma1 (lista de parámetros)
Inicio
< Instrucciones >
Fin
Subprograma2 (lista de parámetros)
Inicio
< Instrucciones >
Fin
Subprograma3 (lista de parámetros)
Inicio
< Instrucciones >
Fin
/* Programa principal */

Inicio

Subprograma1 (lista de parámetros actuales)
Subprograma1 (lista de parámetros actuales)
Subprograma1 (lista de parámetros actuales)

Fin

Variable Global: Se definan antes de los subprogramas, su utilidad radica que si Ud. Desea puede usarlo en cualquier subprograma, pero si obligadamente en el Programa principal.

Variable Local: Solo se definen dentro del subprograma y por lo tanto pierden su valor en otro subprograma.

Aplicación.- Diseñar un diagrama de Flujo que permita crear 3 subprogramas:

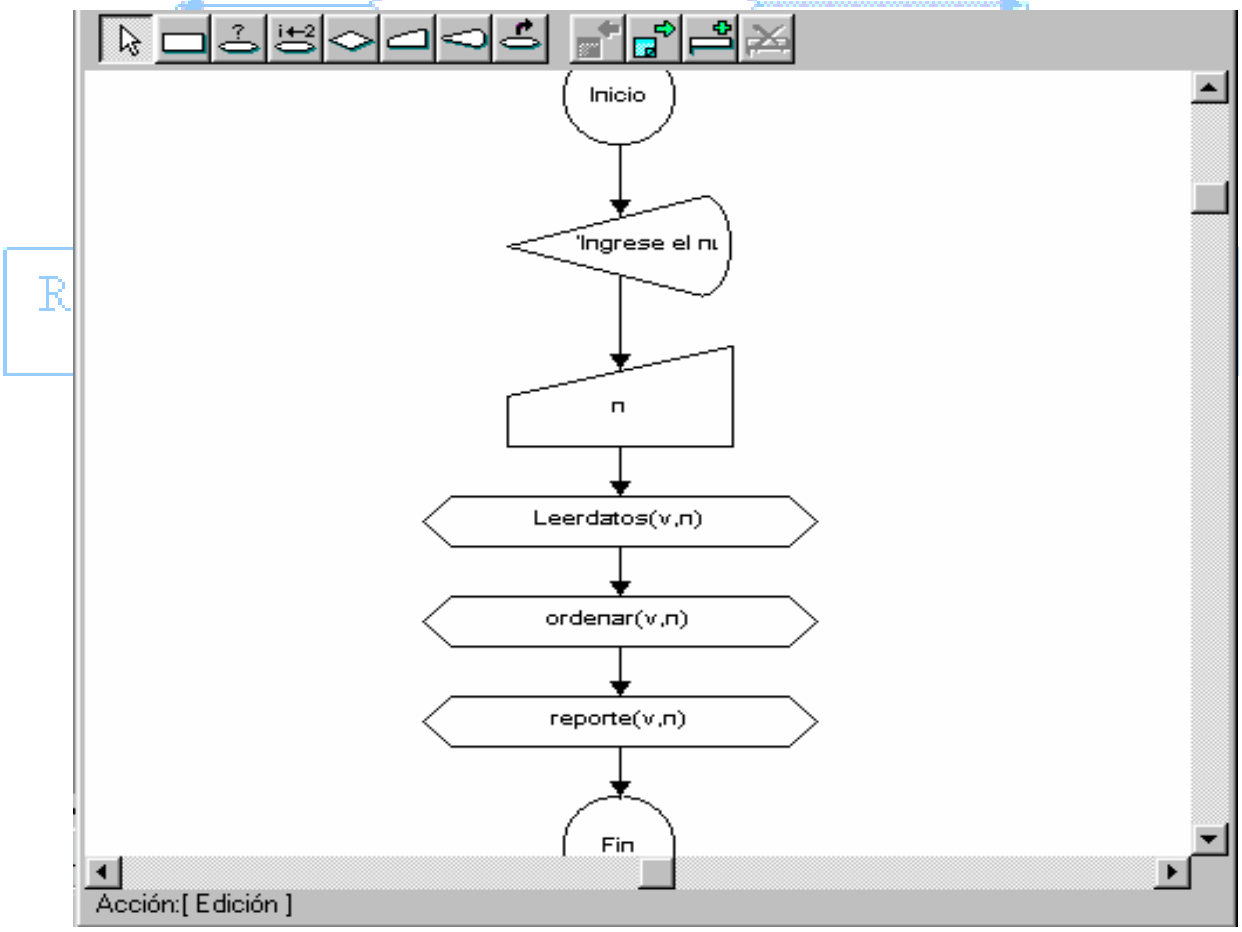
Leerdatos () : Permite leer n elementos de tipo entero.

Ordenar () : Permite ordenar los elemento del vector en forma ascendente.

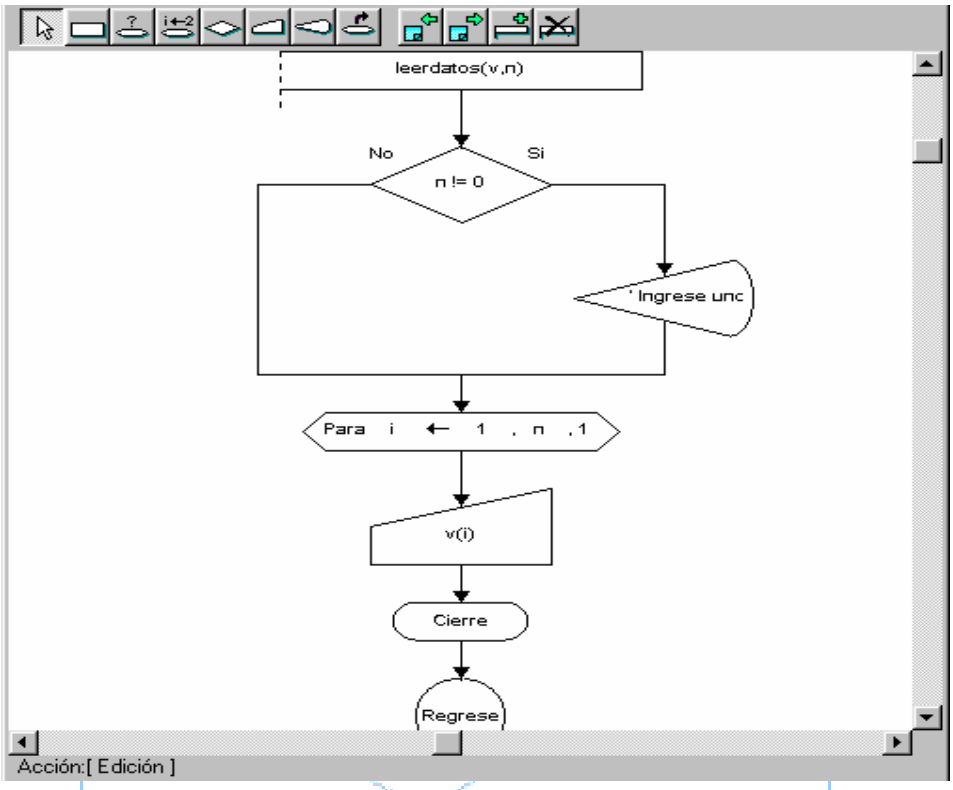
Reporte() : Permite hacer un listado de los elementos y en forma ordenada.

Implementación:

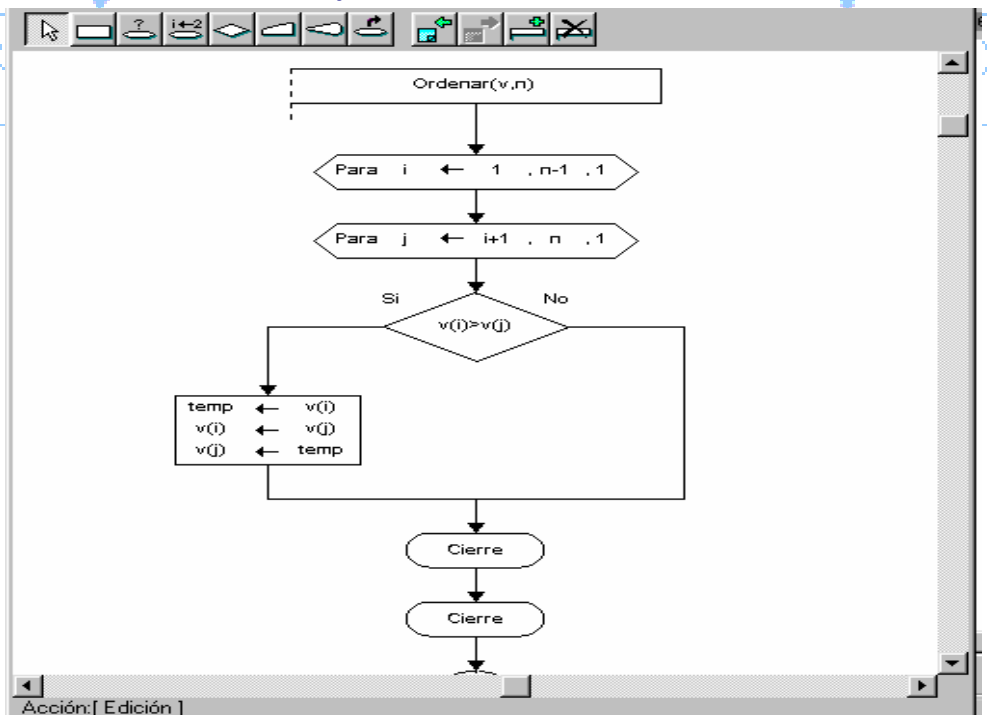
Paso 1.- En la siguiente grafica, se ilustra el diseño de los 3 subprogramas cada uno usa lista de parámetros: v, n



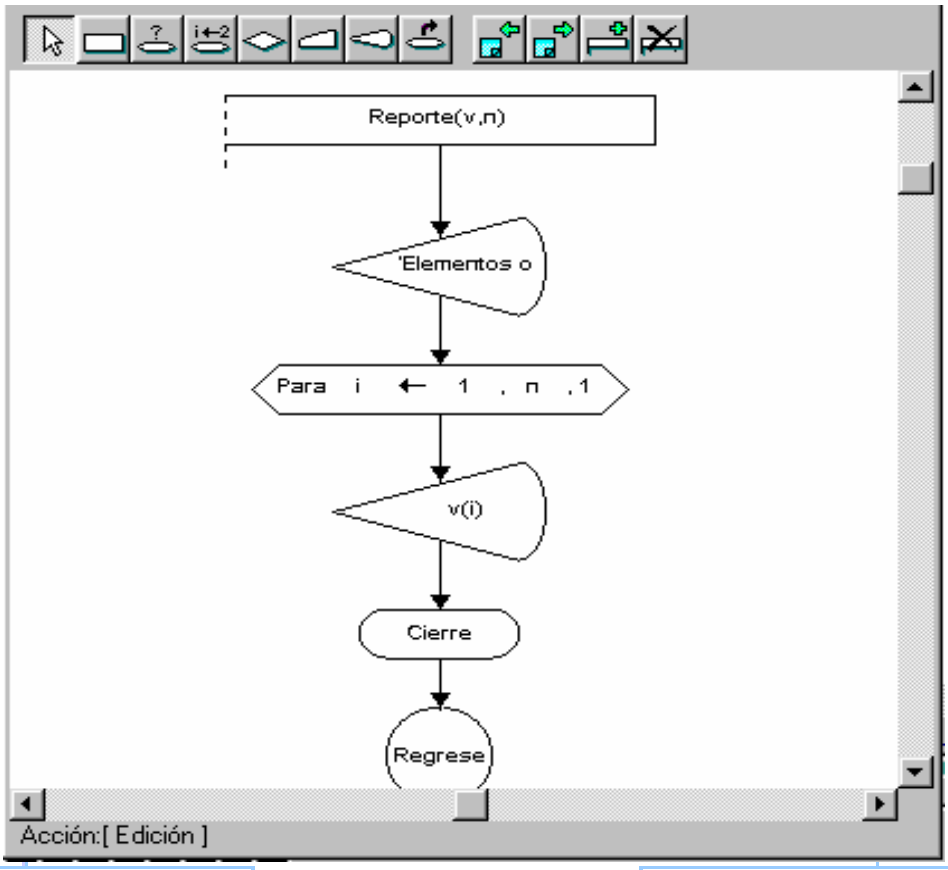
Paso 2.- Diseño del subprograma Leerdatos(v,n): realizar la lectura de n elementos del vector, n es ingresado por el usuario.



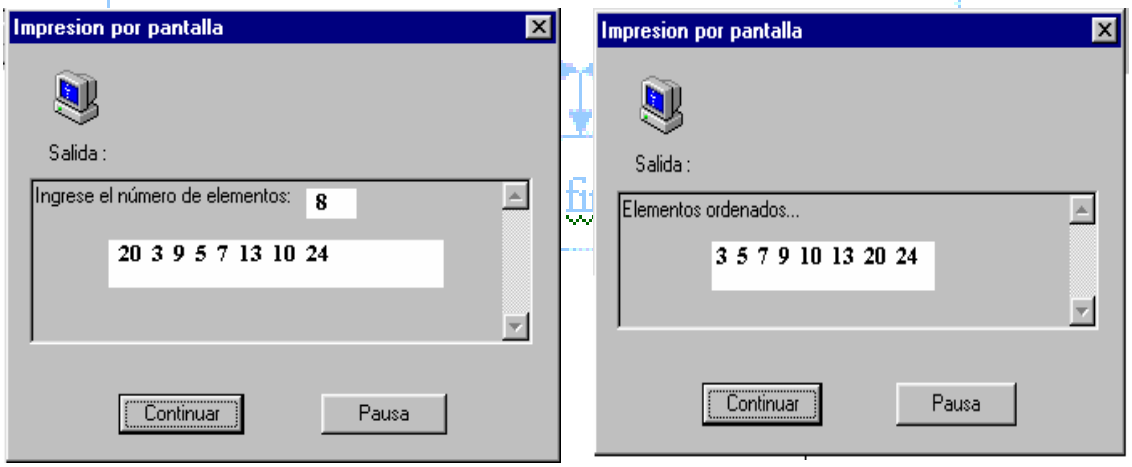
Paso 3.- Diseño del subprograma Ordenar (v,n): realizar la ordenación de datos usando la técnica de la Burbuja.

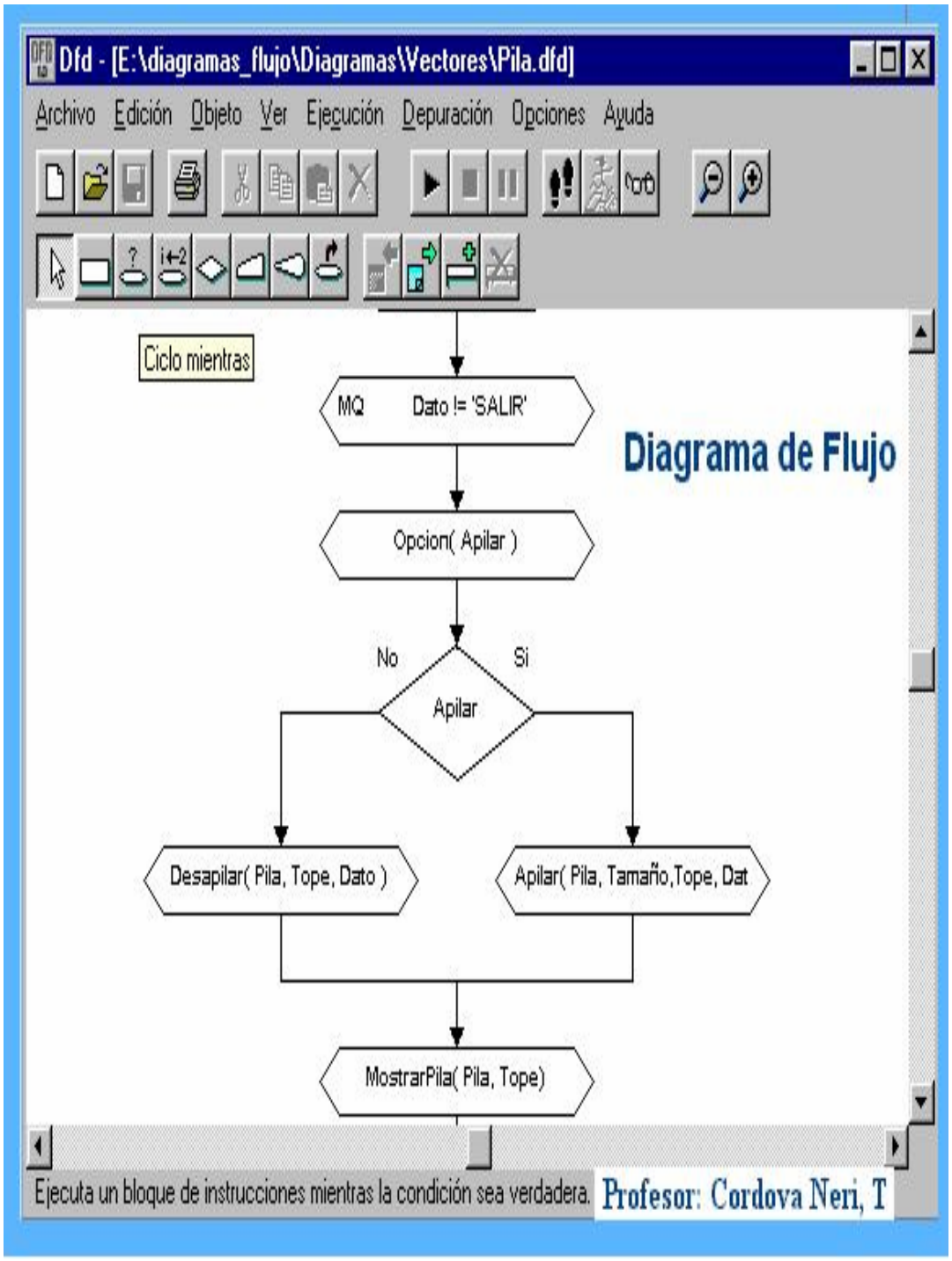


Paso 4.- Diseño del subprograma Reporte (v,n): realizar el reporte de los elementos del vector en forma ordenada.

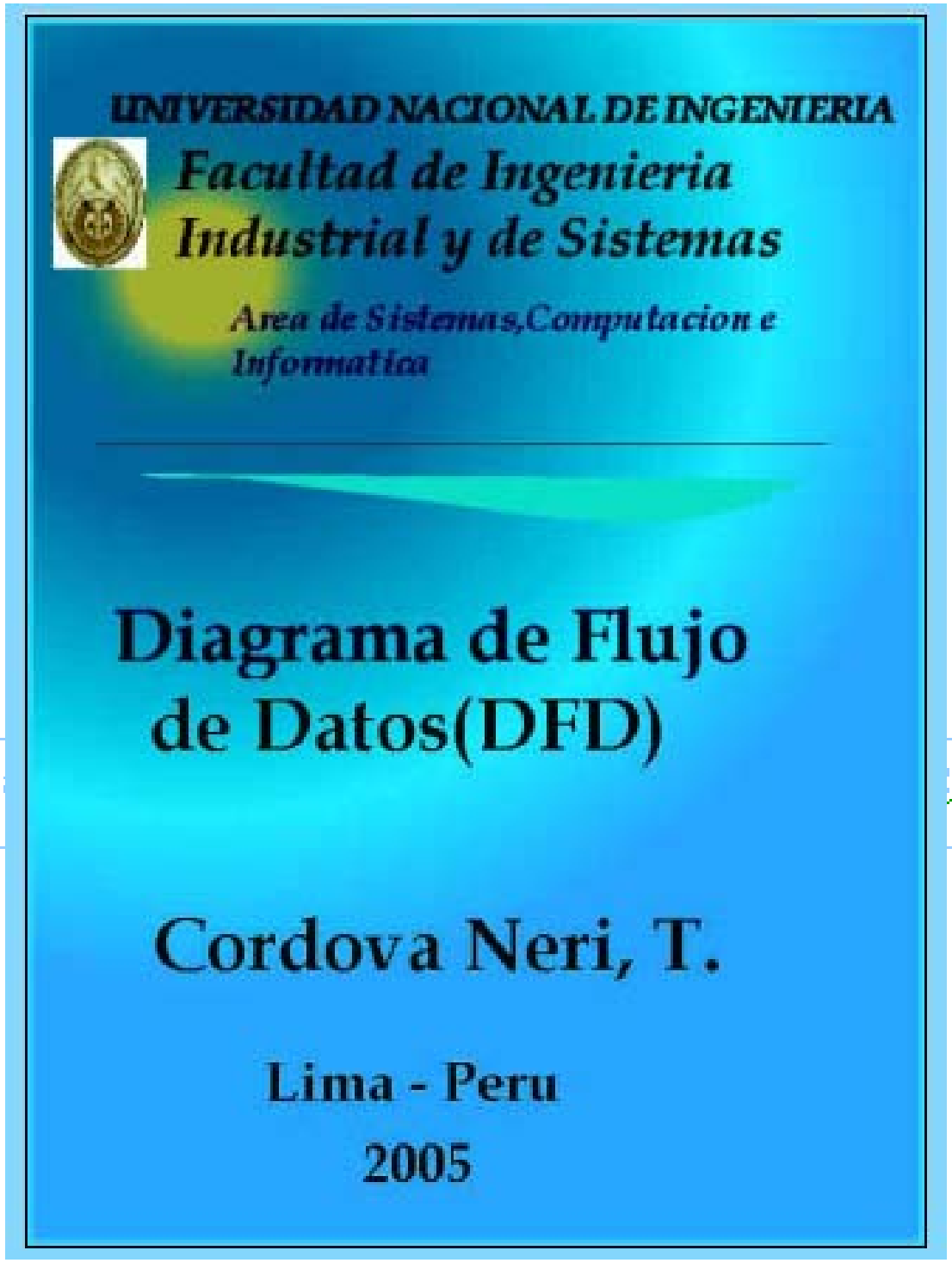


Ejecución.- Al ejecutar, el programa le solicita que ingrese el número de elementos del vector. N=8




Dfd - [E:\diagramas_flujo\Diagramas\Vectores\Pila.dfd]
Archivo Edición Objeto Ver Ejecución Depuración Opciones Ayuda
Ciclo mientras
MQ Dato != 'SALIR'
Diagrama de Flujo
Opcion(Apilar)
No
Si
Apilar
Desapilar(Pila, Tope, Dato)
Apilar(Pila, Tamaño,Tope, Dat
MostrarPila(Pila, Tope)
Ejecuta un bloque de instrucciones mientras la condición sea verdadera.
Profesor: Cordova Neri, T

Lima – Perú


UNIVERSIDAD NACIONAL DE INGENIERIA
Facultad de Ingenieria
Industrial y de Sistemas
Area de Sistemas,Computacion e
Informatica
Diagrama de Flujo
de Datos(DFD)
Cordova Neri, T.
Lima - Peru
2005