UNIVERSIDAD DE LOS ANDES
MERIDA VENEZUELA

Unidad 2. La lógica de programación

Tema 4. Arreglos y estructuras de repetición

UNIVERSIDAD DE LOS ANDES

Arreglos y estructuras de repetición

- Contenido:
  - Introducción
  - Arreglos
    - Vectores
    - Matrices
    - De más de 2 dimensiones
  - Estructura de repetición
    - Repita mientras
    - Repita hasta o hacer mientras
  - Representación algorítmica
  - Codificación
- Objetivo:

  Desarrollar habilidades en el uso de vectores, matrices y las estructuras de repetición
- Bibliografía:
  - Deitel y Deitel, cap. 4 y 6.
  - http://www.ing.ula.ve/~ibc/pr1
  - http://www.webdelprofesor.ula.ve/ingenieria/ibc
  - Navas y Besembel, tema V.
  - Joyanes, sec. 4.6, 4.8 y cap. 6.

UNIVERSIDAD DE LOS ANDES

Tipos y estructura de un arreglo

- Tipos de arreglos:
  - Vector (arreglo unidimensional / 1D), un (1) subíndice
  - Matriz (arreglo bidimensional / 2D), dos (2) subíndices
  - Multidimensional (tres / 3D o más dimensiones), tres (3) o más subíndices
- Estructura de un arreglo:
  - Tipo de dato del arreglo
  - Nombre de referencia que indica la dirección base de memoria de inicio del arreglo
  - Número máximo (esperado o efectivo) de elementos que contendrá el arreglo

Tipo nombreDelArreglo[tamañoMáximo];

Ejemplo: Almacenar cuatro (4) números enteros int num1, num2, num3, num4;

con una matriz de 2x2 int num[2][2];

num

num11	num12
num21	num22

UNIVERSIDAD DE LOS ANDES

Operaciones sobre arreglos

- Leer el valor de un elemento del arreglo

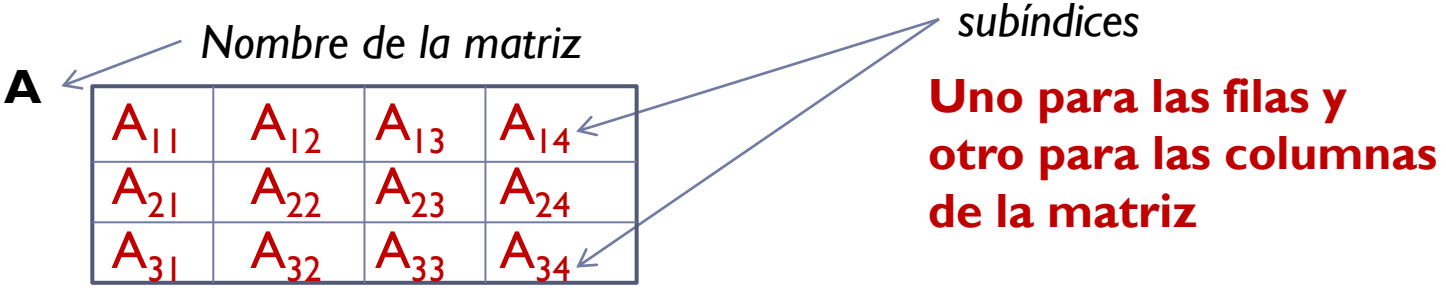
  nombreDelArreglo[posiciónDelElemento]

- Escribir el valor de un elemento en el arreglo

  nombreDelArreglo[posiciónDelElemento]=valor

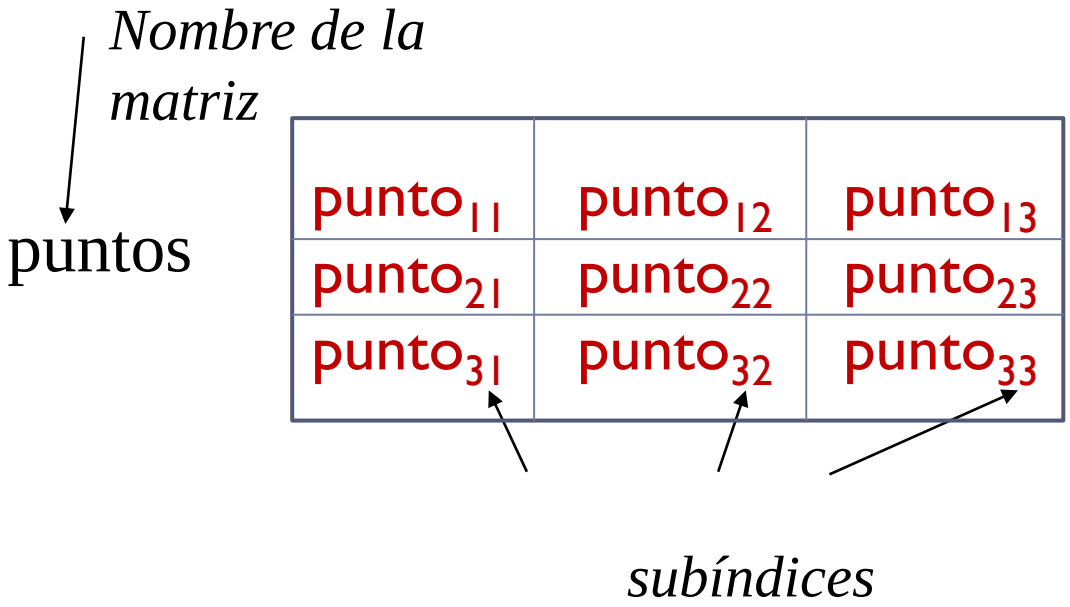
Matriz(Arreglo bidimensional / 2D)

- Grupo de localidades consecutivas de memoria relacionadas por su nombre de referencia y su tipo de dato
- Cada localidad o grupo de localidades almacena un elemento de la matriz



Notación algorítmica de la matriz

- Cada elemento de la matriz es accedido mediante el nombre de la misma y dos subíndices que representan la posición numérica (entero no negativo) de dicho elemento dentro de la matriz



- Crear una matriz de elementos

TipoDeDato nombreDeLaMatriz[numDeFilas][numDeColumnas]

Real puntos[3][3]

UNIVERSIDAD DE LOS ANDES

TAD Matriz

30/11/11 Especificación (7) Matriz[DF][DC]De TipoDeDato		
1	Sintáctica: creaMatriz(Entero+, Entero+,TipoDeDato) -> Matriz[DF][DC], leeElemento(Matriz[DF][DC], Entero+, Entero+) -> TipoDeDato, escribeElemento(Matriz[DF][DC], Entero+, Entero+, TipoDeDato) -> Matriz[DF][DC],	creaMatriz(): Crea un arreglo bidimensional con el número de filas, columnas y tipo de dato suministrados por los parámetros, sin limpiar la memoria. leeElemento(): Devuelve el elemento que está en la posición solicitada en el parámetro. escribeElemento(): Escribe el elemento enviado por el parámetro en la posición indicada, actualizando la matriz.
2	Declaraciones: f, c, df, dc: Entero+ $(0 \leq f \leq DF)$ $(0 \leq c \leq DC)$ el: TipoDeDato $\cup$ {TipoEleNoDefinido}	
3	Semántica: leeElemento(creaMatriz(DF, DC, TipoDeDato), f, c) = TipoEleNoDefinido leeElemento(escribeElemento(creaMatriz(DF, DC, TipoDeDato), f, c, el), f, c) = el leeElemento(escribeElemento(Matriz[DF][DC], f, c, el), f, c) = el	

Declaración de una matriz en C/C++

tipoDeDato nombreDeLaMatriz[numFilas][numColumnas];

- Ejemplos:

```
float   notas[6][4];     // Matriz de 6 filas de 4 notas reales
char    ced[4][10];      // Matriz de 4 filas de cedulas de 10 caracteres
int     w[100][3];       // Matriz de 100 filas de 3 números enteros
int     x[12][30], z[90];    // Una matriz y un vector de enteros
```

Codificación C/C++

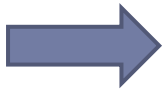
```
#define NUMDIAS  31
#define  NUMHORAS  24

int    mes[NUMHORAS][NUMDIAS];
```

Dibuje la matriz mes

Acceso a los elementos de una matriz

notas

11	12
4	13



Primer elemento: notas[1][1]=11
Segundo elemento: notas[1][2]=12
Tercer elemento: notas[2][1]=4
Sexto elemento: notas[2][2]=13

Cada elemento de una matriz puede usarse como una variable cualquiera

Ejemplos:

```
Promedio[1]=(notas[1][1]+ notas[1][2])/2;
printf("valor= %i", notas[2][2]);
cin>>notas[2][1];
cout<<" Nota promedio del estudiante 2= "<<Promedio[2]<<endl;
```

Iniciación de los elementos de una matriz

Todos los elementos de la matriz tienen asignado valores iniciales

```
int     matriz[7][4] = {0};
    matriz[0][0]=matriz[0][1]=matriz[0][2]= … =matriz[6][3]=0
float   x[5][2]={0.34, 0.45, 0.67, 0.89};
    x[0][0]=0.34   x[0][1]=0.45   x[1][0]=0.67   x[1][1]=0.89
```

No todos los elementos tienen valor inicial

Programación estructurada

- Enfoque disciplinado que permite escribir programas estructurados, utilizando las siguientes tres estructuras de control:
  - Secuencial (asignación, lectura, escritura)
  - Decisión o selección (simple, doble, múltiple)
  - Repetición
    - Repita para
    - Repita mientras
    - Repita hasta

Dahl, Dijkstra y Hoare. Structured Programming. Academic Press, 1972

Dahl 

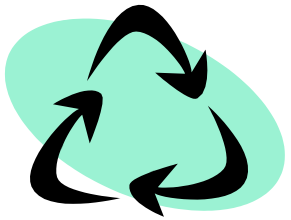
Dijkstra 

Hoare 

UNIVERSIDAD DE LOS ANDES

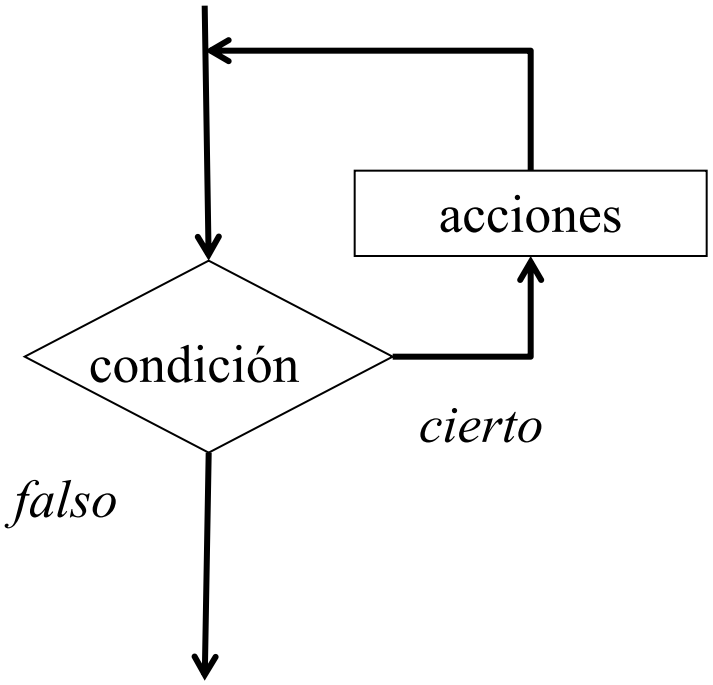
Estructuras de repetición

- Permiten que un conjunto de sentencias (una o varias) de un programa sea ejecutado por la computadora en forma repetida
- Tipos de estructuras de repetición
  - Repetición controlada por un contador:
    - Repita para
  - **Repetición condicional o controlada por un centinela:**
    - **Repita mientras (verificación al inicio del bloque)**
    - Repita hasta (verificación al final del bloque)



Repita mientras

Diagrama de flujo



REPITA MIENTRAS C SE CUMPLA

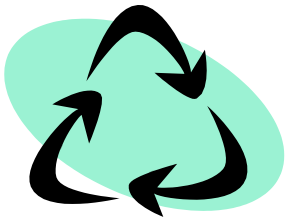
(C) [X1,...,Xn = E1,...,En]

Pseudocódigo en español

```
Repita mientras (<condición>)
        S1
        ....
        Sn
 frm
```

Código en C/C++

```
while (<condición>)
{
        S1
        ....
        Sn
}
```



UNIVERSIDAD DE LOS ANDES

Repita mientras

- Las sentencias (una o mas) del cuerpo del lazo se ejecutan mientras la condición (expresión lógica) es cierta
- Cuando la condición es falsa, termina la ejecución del lazo
- Se pregunta al principio por la condición, por tanto el lazo se ejecuta cero (si la primera vez la condición es falsa) o mas veces
- Si la condición nunca se hace falsa, el programa entra en un "lazo infinito", es decir, las sentencias del cuerpo del lazo se ejecutarán indefinidamente

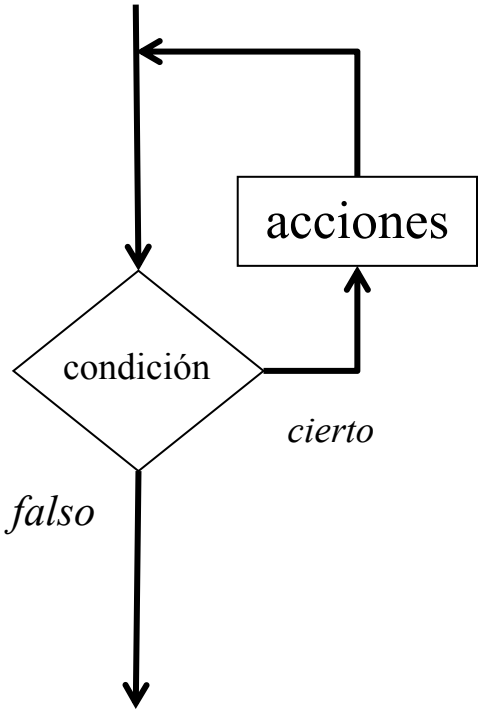
REPITA MIENTRAS C SE CUMPLA

(C) [X1,...,Xn = E1,...,En]

Repita mientras

- INICIAR las variables que intervienen en la condición antes de ejecutar el lazo por primera vez, ya que lo **primero** que hace la estructura es **evaluar la condición**

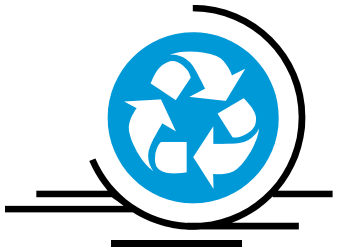
- MODIFICAR dentro del cuerpo del lazo los valores de las variables que intervienen en la condición, para garantizar que en algún momento ésta se haga falsa y el lazo pueda terminar su ejecución, y así garantizar que el lazo no es infinito



UNIVERSIDAD DE LOS ANDES

Acumulador

- Variable cuyo valor se incrementa o decrementa en una cantidad variable cada vez que se produce un determinado suceso o acción
- Se debe realizar primero una operación de iniciación y posteriormente los correspondientes incrementos o decrementos
- Iniciación: Nombre del acumulador = valor inicial
- Acumulación:
  - NombreDelAcumulador = NombreDelAcumulador + valor
  - NombreDelAcumulador = NombreDelAcumulador * valor
  - NombreDelAcumulador = NombreDelAcumulador - valor
  - NombreDelAcumulador = NombreDelAcumulador / valor



UNIVERSIDAD DE LOS ANDES

Acumulador

Iniciación	Acumular (incrementos)	Acumular (decrementos)
totalNota1 = 0 **totalNota2 = 1**	**totalNota1 = totalNota1 + nota** **totalNota2 = totalNota2 * nota**	**totalNota1 = totalNota1 - nota** **totalNota2 = totalNota2 /nota**

Algoritmo

```
MAX, i = 4, 1
(i < MAX) [desplegar i
                i = i+1]
```

```
MAX=4
i=1
Repita mientras (i < MAX)
    Escribir (i)
    i=i+1
Frm
```

Codificación C/C++

```
MAX=4;
i=1;            // inicio de acumulador
while (i < MAX)
{       printf("%i", i);
        i++;        //  acumulación
}
```

Escriba el código equivalente con repita para

UNIVERSIDAD DE LOS ANDES

Repita mientras

Algoritmo	Corrida en frío
MAX, i = 4, 1 (i < MAX) [desplegar i i = i+1] **#include<stdio.h>** **#define MAX 4** **int main()** **{ int i** **for(i=1; i<=MAX; i++)** **printf("%i\n", i);** **return i;** **}**	MAX = 4 i = 1 (1 < 4) => cierto **1** i = 2 (2 < 4) => cierto **2** i = 3 (3 < 4) => cierto **3** i = 4 (4 < 4) => falso

Finaliza el lazo

UNIVERSIDAD DE LOS ANDES

Centinela

- Valor dado a una variable o conjunto de variables que permite detectar cuando se desea terminar de repetir las acciones que constituyen el cuerpo de una estructura de repetición o lazo
- Ejemplo

Iniciar

    bandera = cierto

Cuando se necesite que el lazo termine su ejecución

    bandera = falso

NOTA: Cuando hay mas de una sentencia asociada al **while** se escribe **{ } en C/C++**.

UNIVERSIDAD DE LOS ANDES

Ejemplo 1

Pseudocódigo en Español	Código en C/C++
a, b = 1, 45 (a ≤ b) [a = valor suministrado]	a = 1, b = 45; while(a <= b) cin >> a;

(nota > 15) [cont=cont+1
despliegue
"Estudiante eximido"]
	while(nota > 15)
	{ cont++;
	cout << "Estudiante
	eximido\n";
	}

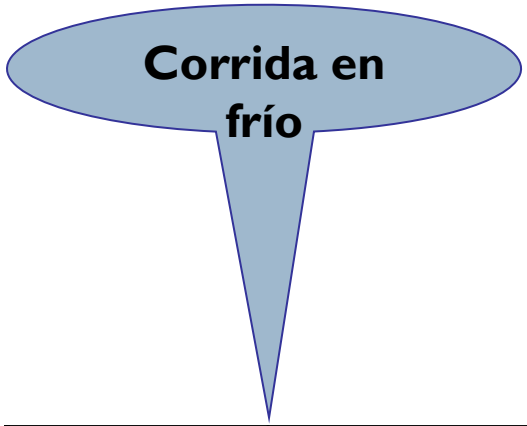
Se modifican las variables de la condición del lazo para provocar su terminación

nota NO fue iniciada => *lazo infinito*

UNIVERSIDAD DE LOS ANDES

Ejemplo 2

```
#include <iostream.h>

void main ()

{   int i = 1;

    while ( i < 3 )

    {  cout <<  "i es menor que 3" << endl;

        i++;

    }

    cout << "Se terminó el lazo" << endl ;

}
```



iteración	i
(0)	1
(1)	2
(2)	3

UNIVERSIDAD DE LOS ANDES

Ejercicio resuelto 1

- Enunciado del problema

  Sumar todos los enteros pares desde 2 hasta 100

- Análisis E-P-S

Entrada	Variable	Descripción	Tipo de dato	Rango válido
	N/A			
Proceso	para cada numero par generado entre 2 y 100 acumular su valor en el acumulador de pares (ap)			
Salida	Mensaje indicando ap	Acumulador de pares	Entero	>0

Ejercicio resuelto 1. Diseño

sumaParesRM		
	{pre: }	{pos: AP > 0}
1 2 3	AP, num=0, 2 (num < 100)[AP, num = AP + num, num + 2] Desplegar "Suma de los números pares entre 2 y 100 = ", AP	•**AP**: Entero+. Acumulador de números pares •**num:** Entero+. Contador de números pares
1	-> Suma de los números pares entre 2 y 100 = 2550	Caso exitoso

Completar la tabla

num	AP	num	AP	num	AP	num	AP	num	AP	num	AP	num	AP	num	AP
	0	14	56	28	210	42	462	56		70		84		98	
2	2	16	72	30	240	44	506	58		72		86		100	2250
4	6	18	90	32	272	46	552	60		74		88			
6	12	20	110	34	306	48	600	62		76		90			
8	20	22	132	36	342	50	650	64		78		92			
10	30	24	156	38	380	52	702	66		80		94			
12	42	26	182	40	420	54	756	68		82		96			

Ejercicio resuelto 1. Implementación

```
#include <stdio.h>
int main()
{
   unsigned int          AP = 0, num=2;
   while ( num <= 100)
   {      AP += num;
          num +=2;
   }
   printf("Suma de numeros pares entre 2 y 100 = %d\n", AP);
   return 0;
}
```

UNIVERSIDAD DE LOS ANDES

Resumen

Tipo nombreDelArreglo[tamañoMáximo]= {listaValoresIniciales};

nombreDelArreglo[posiciónDelElemento]

nombreDelArreglo[posiciónDelElemento]=valor

Código en C/C++

```
while (<condición>)
{
        S1
        ....
        Sn
}
```

REPITA MIENTRAS C SE CUMPLA

(C) [X1,...,Xn = E1,...,En]

UNIVERSIDAD DE LOS ANDES

Ejercicio resuelto 2

- Enunciado del problema

  Dadas las notas de n estudiantes de las materias de Programación 1, Física11 y Cálculo 10. Calcular el promedio de notas de cada estudiante

- Análisis E-P-S

Entrada	Variable	Descripción	Tipo de dato	Rango válido
	n notas[n][4]	Numero de estudiantes Notas de Programación 1, Física11, Cálculo 10 y promedio	Entero+ MatrizDeReal	≥0 notas[i][j]∈[0, 20]
Proceso	Para cada estudiante hacer: Calcular notas[i][4] = $\frac{notas[i][1]+notas[i][2]+notas[i][3]}{3}$			
Salida	Mensaje indicando el valor de promedio para cada estudiante	Promedio de las tres notas dadas para cada estudiante	VectorDeReal	[0, 20]

Ejercicio resuelto 2. Diseño

promedio3notas

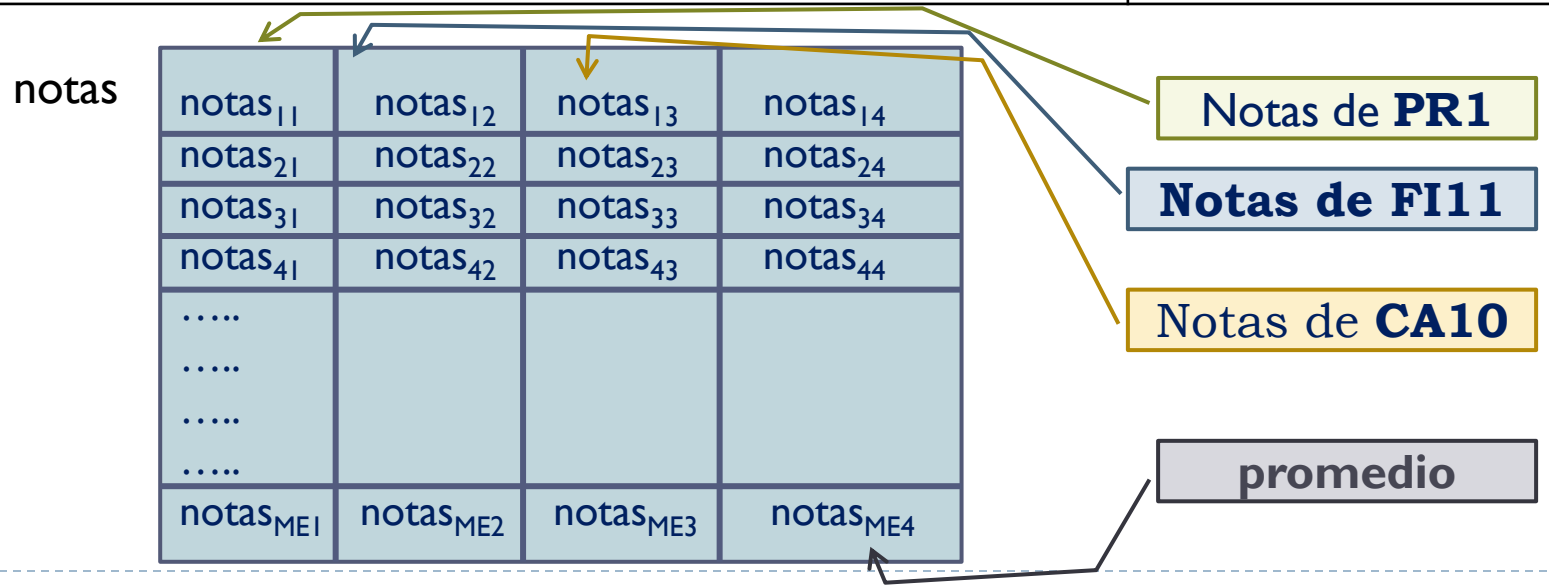
{pre: n ∈ ℵ} {pos: n∈ ℵ, ∀i, $PR1_i$, $FI11_i$, $CA10_i$, $promedio_i$ ∈ℜ }

```
1  n = -2
2  (n < 1) [ Desplegar "Introduzca el numero de estudiantes ≥ 1"
                n = valor suministrado  ]
3  [ Desplegar "Para el estudiante ", k
     notas[k][1]= -1
     (notas[k][1] <0 ∨ notas[k][1] > 20) [Desplegar "Introduzca la nota de PR1
entre 0 y 20"
                                                  notas[k][1] = valor suministrado ]
      Si (0≤ notas[k][1]≤ 20)  entonces
        notas[k][2] = -1
        (notas[k][2] < 0 ∨ notas[k][2] > 20) [Desplegar "Introduzca la nota de
Fisica1 entre 0 y 20"
                                                  notas[k][2] = valor suministrado ]
        Si (0≤ notas[k][2]≤ 20) entonces
          notas[k][3] = -1
          (notas[k][3] < 0 ∨ notas[k][3] > 20) [Desplegar "Introduzca la nota de
Cálculo 1 entre 0 y 20"
                                                  notas[k][3] = valor suministrado ]
           Si (0≤ notas[k][3]≤ 20)
              Desplegar "Promedio de notas del estudiante ", k, " = ", notas[k][4]=
(notas[k][1]+notas[k][2]+notas[k][3])/3
           fsi
```

- **n:** Natural. Número total de estudiantes de la asignatura.
- **k:** Natural. Contador de estudiantes.
- notas: MatrizDeReal[n][4]. Notas de las materias Programación 1, Física11, Cálculo 10 y su promedio para cada estudiante. Cada nota debe estar en el rango de 0 a 20.

Ejercicio resuelto 2. Diseño (cont)

	promedio3notas {pre: n ∈ ℵ} {pos: n∈ ℵ, ∀i, $PR1_i$, $FI11_i$, $CA10_i$, $promedio_i$ ∈ℜ }	
	fsi fsi] k = 1, n, 1	**Compare esta solución con la de la clase 6 para el mismo enunciado**
1	n = 1, notas[1][1]=12, notas[1][2]=10, notas[1][3]=11 => Promedio de notas del estudiante 1=11	Caso límite
2	n = 2, notas[1][1]=11, notas[1][2]=11, notas[1][3]=11 => Promedio de notas del estudiante 1=11 notas[2][1]=1, notas[2][2]=1, notas[2][3]=1 => Promedio de notas del estudiante 2=1	Caso exitoso



UNIVERSIDAD DE LOS ANDES

Codificación en C++

```
#include <iostream>
using namespace std;
#define  ME  50
int    main ( )
{      int n = -2, k;
       float       notas[ME][4];
       while(n < 0 || n > ME)
       {      cout << "Introduzca el numero de estudiantes, entero mayor que 0 y menor que \n"<< ME;
              cin >> n;
       }
       for(k=1; k<=n; k++)
       {      cout << "Introduzca para el estudiante"<<k << endl;
              notas[k][0] = -1.0;
              while(notas[k][0] < 0.0 || notas[k][0] > 20.0)
              {      cout<<"Introduzca la nota de Programamcion 1"<<endl;
                     cin >> notas[k][0];
              }
              if(notas[k][0] < 20.0 && notas[k] [0]> 0.0)
                     notas[k][1] = -1.0;
                     while(notas[k][1] < 0.0 || notas[k][1] > 20.0)
                     {      cout << "Introduzca la nota de Fisica1" << endl;
                            cin >> notas[k][1];
                     }
```

UNIVERSIDAD DE LOS ANDES

Ejercicio resuelto 2

```
        if(notas[k][1] < 20.0 && notas [k][1] > 20.0)
            notas[k][2] = -1.0;
            while(notas[k][2] < 0.0 || notas[k][2] > 20.0)
            {    cout << "Introduzca la nota de Calculo 1" << endl;
                    cin >> notas[k][2];
            }
             if(notas[k][2] < 20.0 && notas[k][2] > 0.0)
            {    notas[k][3] = (notas[k][0]+notas[k][1]+notas[k][2])/3.0;
                cout << "Promedio del estudiante " << k << "= " << notas[k][3] <<
    endl;
            }
    }
    return 0;
}
```

Proponga otros diseños junto con sus implementaciones al mismo problema

- ¿Qué es una matriz o arreglo bidimensional?
- ¿Cómo es el TAD Matriz?
- ¿Cómo se declaran e inician las matrices?
- ¿Qué es una estructura de repetición controlada por un centinela?
- ¿Cómo se define una estructura de repetición controlada por un centinela?
- ¿Cómo se utiliza un acumulador y un centinela en un repita mientras?
- ¿Cómo se prueba en frío un programa con repita mientras?
- ¿Cómo se anidan las estructuras de decisión y de repetición?

Resumen

¿Cuáles son los conceptos relevantes de esta clase?

UNIVERSIDAD DE LOS ANDES

Ejercicios

- **Para cada uno de los enunciados dados a continuación realizar: Análisis en E-P-S, Diseño en pseudocódigo y codificación en C o C++**

1. Encuentre el número de puntos con coordenadas enteras que están dentro de la elipse $6x^2 + 5y^2 = 4$
2. Imprimir la tabla de multiplicar de un número dado hasta que la persona no desee otra tabla. El funcionamiento del programa se muestra en el siguiente ejemplo de ejecución

   Introduzca un número: 5

   La tabla de multiplicar del 5 es:

   5 x 1 = 5

   5 x 2 =10

   ...

   5 x 10 =50

UNIVERSIDAD DE LOS ANDES

Ejercicios

4. Escribir todos los enteros positivos menores que 100 omitiendo aquellos divisibles por 7
5. Leer un número entero n y calcular e imprimir su inverso 1/n. Considerar el caso especial del valor 0, en cuyo caso el programa deberá escribir el mensaje "ERROR -división por cero“ y terminar
6. Ordenar ascendentemente una matriz de diez por seis elementos reales (matriz[10][6]) utilizando la columna **col** de la misma, cuyo valor será indicado desde el teclado.