



UNIVERSIDAD
DE LOS ANDES
MERIDA VENEZUELA

Estructuras de Decisión Simple y Doble

Prof. Judith Barrios Albornoz

Departamento de Computación
Escuela de Ingeniería de Sistemas
Facultad de Ingeniería
Universidad de Los Andes

Semestre A_2013

Estructuras no secuenciales

- **Flujo de control:** Habilidad de **definir el orden** en el cual las sentencias de un programa son ejecutadas

- Secuencia

Los programas que hemos escrito hasta ahora han consistido en una **serie de sentencias que se ejecutan secuencialmente**, una después de la otra (**asignación, lectura y escritura**)

➡ ○ Estructuras de decisión o selección

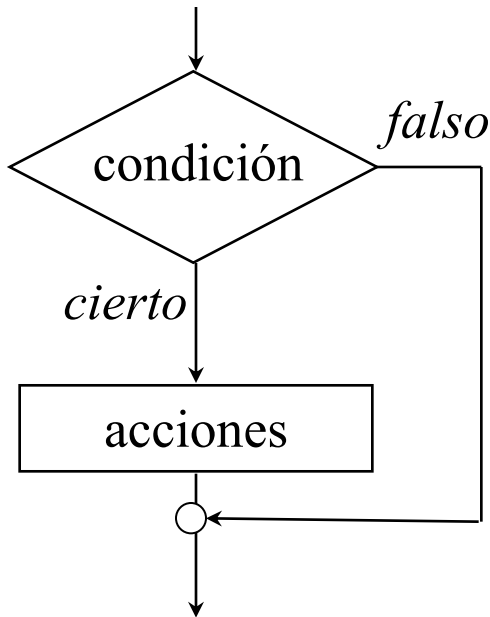
- Estructuras de repetición

Estructuras de Decisión

- Permite **seleccionar la próxima sentencia** a ejecutarse basándose en una **condición** (**expresión lógica o una variable lógica**) que es **evaluada** y, en **función del resultado** (**cierto o falso**) de la evaluación, se **bifurcará** a un determinado punto para continuar la ejecución del programa
- Tipos:
 - Simple
 - Doble
 - Múltiple

Estructura de decisión simple

Diagrama de flujo



- Pseudocódigo en español

Si (*<condición>*) entonces

S_1

....

S_n

fsi

- Código en C/C++

if (*<condición>*)

{

S_1

....

S_n

}

Estructura de Decisión Simple

- Esta estructura evalúa la condición
 - Si la *condición* es **cierta** entonces **ejecuta** el conjunto de **sentencias** definidas por S_i con $i = 1 .. N$
 - Si la *condición* es **falsa** entonces **NO ejecuta** el conjunto de **sentencias** definidas por S_i con $i = 1 .. N$

Estructura de decisión simple

Diseño	Código en C/C++
Si ($x > 0.0$) entonces escribir “x es positivo” fsi	if ($x > 0.0$) printf (“x es positivo\n”);
Si ($a < b$) entonces $b = a + b$ fsi	if ($a < b$) $b = a + b$;

NOTA: Cuando hay una sola sentencia asociada al **if** no se escribe { }

Estructura de decisión simple

Diseño	Código en C/C++
Si (nota > 15) entonces conta=conta+1 escribir “Eximido” fsi	if (nota > 15) { conta++; printf (“Eximido\n”); }
Si (x ≥ 0.0) entonces y = x + y cont=cont+1 fsi	if (x ≥ 0.0) { y += x; cont++; }

NOTA: Cuando hay mas de una sentencia asociada al if se coloca { }

Ejemplo 1

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

```
#define falso 0
```

```
#define cierto 1
```

```
void main ()
```

```
{
```

```
    int miPrimeraLogica, miSegundaLogica;
```

```
    miPrimeraLogica = cierto;
```

```
    miSegundaLogica = !miPrimeraLogica;
```

```
    if ( miSegundaLogica == cierto)
```

```
        printf ("miSegundaLogica debe tener el valor cierto\n");
```

```
    if ( !miSegundaLogica == cierto)
```

```
        printf (" miSegundaLogica debe tener el valor falso\n");
```

```
}
```


Ejemplo 2

Dados tres números enteros, determinar **si la suma de cualquier pareja de ellos es igual al tercer número**. Si se cumple esta condición, escribir “*Iguales*” y, en caso contrario, escribir “*Distintos*”

Análisis E-P-S

Entradas: $A, B, C \in \mathbb{Z}$

Proceso:

- Determinar si $A + B = C \Rightarrow$ “Iguales”
- Determinar si $A + C = B \Rightarrow$ “Iguales”
- Determinar si $B + C = A \Rightarrow$ “Iguales”
- Si no se cumple ninguno de los anteriores \Rightarrow “Distintos”

Salida: Mensaje indicando si la suma de dos números es igual “Iguales” o diferente “Distintos” a un tercero

Diseño del ejemplo 2

Ejemplo2

{pre: ninguna}

{pos: $A, B, C \in \mathbb{Z}$ }

1	Escribir “Introduzca tres numeros enteros ”	<ul style="list-style-type: none"> ● A, B, C: Entero. Valores de entrada para determinar si la suma de cualquier pareja de ellos es igual al tercer número
2	Leer A, B, C	
3	Si $(A+B = C)$ entonces Escribir “Iguales” fsi	
4	Si $(A+C = B)$ entonces Escribir “Iguales” fsi	
5	Si $(B+C = A)$ entonces Escribir “Iguales” Fsi	
6	Si $((A+B \neq C) \wedge (A+C \neq B) \wedge (B+C \neq A))$ entonces Escribir “Distintos” fsi	
1	$A = 0, B = 0, C = 0 \rightarrow$ Iguales	Caso exitoso
2	$A = 4, B = 3, C = 8 \rightarrow$ Distintos	Caso exitoso

Código del ejemplo 2

Codificación

```
#include <stdio.h>

void main ()
{
    int A, B, C;
    printf ("Introduzca tres números enteros\n");
    scanf ("%d%d%d", &A, &B,&C);
    if (A + B == C)
        printf (" Iguales\n");
    if (A + C == B)
        printf (" Iguales\n");
    if (B + C == A)
        printf ("Iguales\n");
    if ((A + B != C) && (A + C != B) && (B + C != A))
        printf ("Distintos\n");
}
```

Anidamiento de estructuras de decisión simples

- Se anidan colocando una en el interior de la otra

Diseño	Código en C/C++
<pre> Si (condición1) entonces Si (condición2) entonces fsi fsi </pre>	<pre> if(condicion1) { if(condicion2) { } } ; </pre>


Anidamiento válido


Anidamiento NO válido

Ejemplo A de anidamiento

Diseño	Código en C/C++
<p>Si (edad < 18) entonces</p> <p> Escribir “Menor de edad”</p> <p> menores=menores+1</p> <p>Si (sexo = ‘F’) entonces</p> <p> Escribir “Sexo femenino”</p> <p> femenino=femenino+1</p> <p>fsi</p> <p>fsi</p>	<pre>if(edad < 18) { printf (“Menor de edad\n”); menores++; if(sexo == ‘F’) { printf (“Sexo femenino\n”); femenino++; } };</pre>

Ejemplo B de anidamiento

Diseño	Código en C/C++
<p>Si (edad < 18) entonces</p> <p> Escribir “Menor de edad”</p> <p> menores=menores+1</p> <p> Si (sexo = ‘F’ o sexo = ‘f’)</p> <p> entonces</p> <p> Escribir “Sexo femenino”</p> <p> femenino=femenino+1</p> <p> fsi</p> <p>fsi</p>	<pre>if(edad < 18) { printf (“Menor de edad\n”); menores++; if(sexo == ‘F’ sexo == ‘f’) { printf (“Sexo femenino\n”); femenino++; } };</pre>

Ejemplo C de anidamiento

Diseño	Código en C/C++
<p>Si (edad < 18) entonces</p> <p> Si (sexo = 'F') entonces</p> <p> Escribir “Sexo femenino”</p> <p> femenino=femenino+1</p> <p> fsi</p> <p>fsi</p>	<pre>if(edad < 18) if(sexo == 'F' sexo == 'f') { printf (“Sexo femenino\n”); femenino++; };</pre>

Ejercicios 1

1. Si $x = 15$ determinar su valor después de la ejecución del siguiente segmento de programa en C:

```
if( x > 0 )  x+=32;
```

2. Si $z = 6.7$ determinar el valor de la variable conta después de la ejecución del siguiente segmento de programa en C:

```
conta = 0;
```

```
neg = 0;
```

```
if (z < 0.0)
```

```
    neg = neg + z;
```

```
conta = neg + 1;
```


Ejercicio 2

3. Si $z = 6.7$ determinar el valor de la variable **conta** después de la ejecución del siguiente segmento de programa en C:

```
conta = 0;  
neg = 0;  
if (z < 0.0)  
{  
    neg = neg + z;  
    conta = neg + 1;  
}
```

```
conta = 0  
neg = 0  
(6.7 < 0.0) falso  
conta = 0
```

Ejercicio 3

4. Determinar el precio de un **pasaje de ida y vuelta en avión** conociendo la distancia a viajar; cuando el número de días de estancia es **superior a 7** y la distancia d es **superior a 800 km.** el precio **por km.** es p Bs

Análisis E-P-S

- **Entrada:**
 - El precio por Km. (p pertenece \mathbb{R}^+)
 - La distancia a viajar (d pertenece \mathbb{R}^+)
 - El número de días de estancia (días pertenece \mathbb{Z}^+)
- **Proceso:**

Si el número de días de estancia es > 7 y la distancia d es > 800 km, el precio es $2 \cdot d \cdot p$
- **Salida:** Precio del pasaje (precio pertenece \mathbb{R}^+)



UNIVERSIDAD
DE LOS ANDES
MÉRIDA VENEZUELA

Ejercicio 3 - continua

precioPasajeAvion

{pre: ninguna}

{pos: precio $\in \mathbb{R}^+$ }

1 Escribir “Introduzca la distancia en km. a viajar ”
2 Leer **d**
3 Escribir “Introduzca el precio por km.”
4 Leer **p**
5 Escribir “Introduzca cuantos días de estadía”
6 Leer **días**
7 Si $((\mathbf{días} > 7) \wedge (\mathbf{d} > 800))$ entonces
 precio = $2\mathbf{dp}$
 Escribir “Precio a pagar “, **precio**
8 **Fsi (7)**

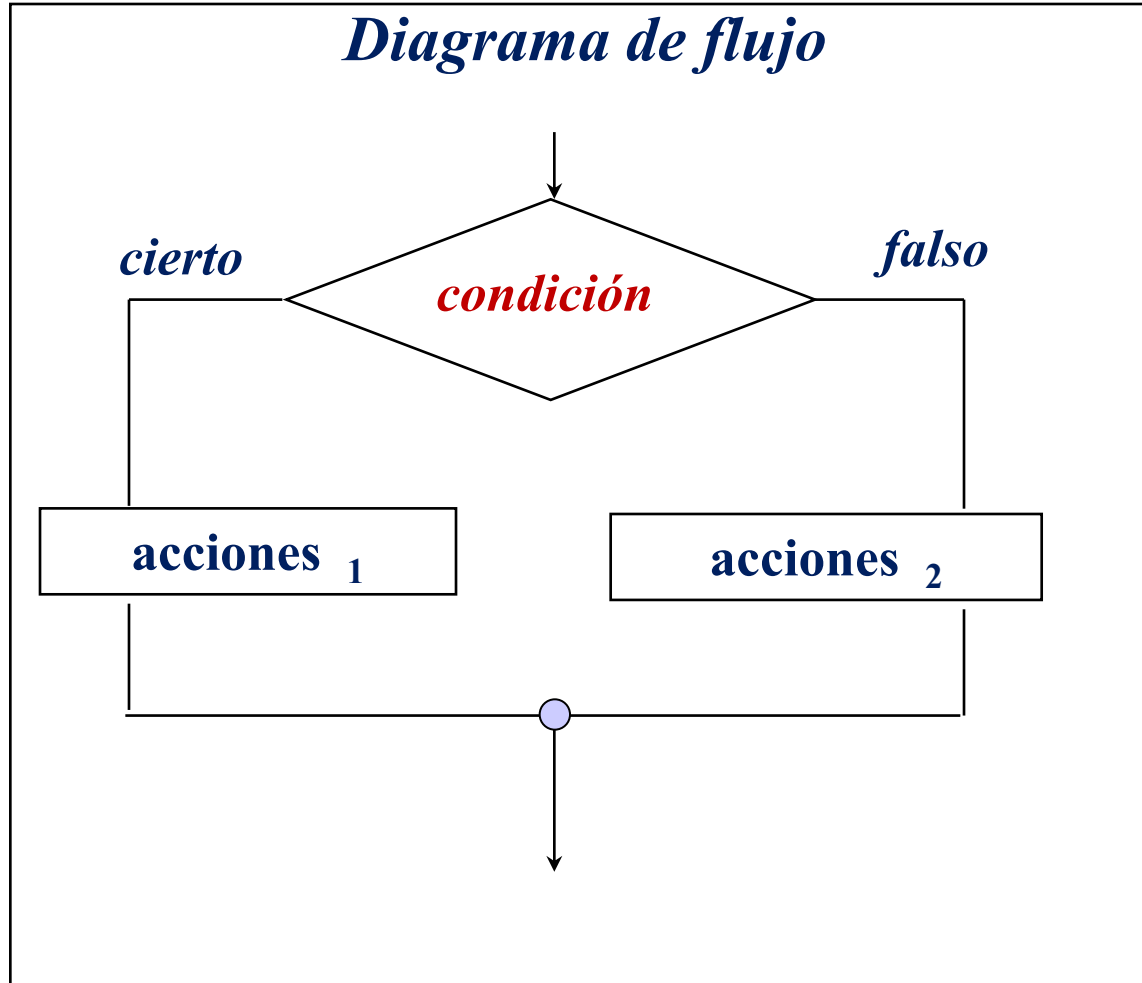
● **d:** *Real*⁺. Distancia en km. a viajar
● **p:** *Real*⁺. Precio por kilometro recorrido
● **días:** *Entero*⁺. Número de días de estadía en el lugar
● **precio:** *Real*⁺. Precio en bolívares del pasaje ida y vuelta en avión

1 $d = 0.0, p = 0.0, \text{días} = 0 \text{ precio} = 0.0$
2 $d = 450.0, p = 500.0, \text{días} = 8 \rightarrow \text{precio} = 225000.00$
3 $d = 950.0, p = 300.0, \text{días} = 8 \rightarrow \text{precio} = 570000.00$

Caso no exitoso
Caso no exitoso
Caso exitoso

Estructura de Decisión Doble

Diagrama de flujo



Estructura de Decisión Doble

Pseudocódigo en español

Si (<condición>) entonces

S_{11}

....

S_{1n}

sino

S_{21}

....

S_{2m}

fsi

Código en C/C++

if (<condición>)

{ S_{11}

....

S_{1n}

}

else

{ S_{21}

....

S_{2m}

}

Estructura de Decisión Doble

- Permite elegir entre dos opciones o alternativas posibles, en función del cumplimiento o no de la *condición*:
 - Si el resultado de evaluar la *condición* es **CIERTO** se ejecutará un **determinado grupo** de sentencias S_{1i} ,
 - $i = 1, \dots, n$.
 - Si el resultado de evaluar la *condición* es **FALSO** se ejecutará otro **grupo determinado** de sentencias S_{2k} ,
 - $k = 1, \dots, m$

Estructura de Decisión Doble

Pseudocódigo en Español	Código en C/C++
Si (nota \geq 10) entonces Escribir “Aprobado” sino Escribir “Reprobado” fsi	if (nota \geq 10) printf (“Aprobado\n”) else printf (“Reprobado\n”);
Si (nota \geq 10) entonces notaMas=notaMas+nota Escribir “Aprobado” sino Escribir “Reprobado” fsi	if (nota \geq 10) { notaMas+=nota; printf (“Aprobado\n”) } else printf (“Reprobado\n”);

Estructura de decisión doble

Pseudocódigo en Español	Código en C/C++
Si (nota \geq 10) entonces Escribir “Aprobado” sino reprobado=reprobado+1 Escribir “Reprobado” fsi	if (nota \geq 10) printf (“Aprobado\n”) else { printf (“Reprobado\n”); reprobado++; };
Si (nota \geq 10) entonces aprobado=aprobado+1 Escribir “Aprobado” sino reprobado=reprobado+1 Escribir “Reprobado” fsi	if (nota \geq 10) { aprobado++; printf (“Aprobado\n”); } else { reprobado++; printf (“Reprobado\n”) };

Ejemplo 3

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

```
void main ( )
```

```
{
```

```
    int miEntero;
```

```
    printf (“Introduzca un numero entero\n”);
```

```
    scanf (“%d”, &miEntero);
```

```
    if ( miEntero == 0 )
```

```
        printf ("miEntero es igual a cero\n");
```

```
    else
```

```
        printf (“miEntero no es igual a cero\n”);
```

```
    printf (“termino el programa\n”);
```

```
}
```

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

```
void main ()
```

```
{
```

```
    int m;
```

```
    printf ("Introduzca un numero entero\n");
```

```
    scanf ("%d", &m);
```

```
    if ( (m >= 1) && (m <= 10) )
```

```
        printf ("%d esta entre 1 y 10", m);
```

```
    else
```

```
        printf ("%d no esta entre 1 y 10", m);
```

```
}
```

Ejemplo 5

Calcular el salario semanal de un empleado, sabiendo que éste se calcula en base a las horas semanales trabajadas y de acuerdo a un precio especificado por hora. Si se pasa de las 40 horas semanales, las horas extras se pagarán a razón de 1,5 veces la hora ordinaria

Análisis E-P-S

Entradas: Horas semanales trabajadas ($HST \in \mathbb{R}^+$)

Precio por hora ($PH \in \mathbb{R}^+$)

Proceso: - Si horas semanales trabajadas (HST) ≤ 40

Calcular salario semanal (SS) = $HST * PH$

- Si horas semanales trabajadas (HST) > 40

Calcular salario semanal

$$(SS) = 40 * PH + PH * 1,5 * (HST - 40)$$

Salida: salario semanal ($SS \in \mathbb{R}^+$)

Diseño ejemplo 5

salarioMensual

{pre: ninguna}

{pos: $SS \in \mathbb{R}^+$ }

1	Escribir “Introduzca el numero de horas trabajadas”	-HST: <i>Real</i> ⁺ . Horas semanales trabajadas -PH: <i>Real</i> ⁺ . Precio por horas -SS: <i>Real</i> ⁺ . Sueldo semanal
2	Leer HST	
3	Escribir “Introduzca el precio por hora “	
4	Leer PH	
5	Si ($HST \leq 40$) entonces $SS = HST * PH$ sino $SS = 40 * PH + PH * 1,5 * (HST - 40)$ fsi	
6	Escribir “Sueldo semanal= “, SS	
1	$HST = 44.0, PH = 850.0 \rightarrow SS = 39100.0$	Caso exitoso
2	$HST = 0.0, PH = 950.0 \rightarrow SS = 0.0$	Caso exitoso
3	$HST = 47.0, PH = 0.0 \rightarrow SS = 0.0$	Caso exitoso

Ejemplo 6

Realizar A (E-P-S), diseño y codificación de un programa que **lea un número entero y compruebe si es correcto.**

En caso de que sea correcto, el programa imprimirá el mensaje “CORRECTO”, si no es correcto, el programa imprimirá el mensaje "ERROR - número incorrecto“.

Considere los siguientes casos para decidir si un número es o no correcto:

a) El número es correcto si se **encuentra en el rango** definido por dos valores constantes MINIMO y MAXIMO.

$$(\text{MINIMO} \leq \text{num} \leq \text{MAXIMO})$$

b) El número es correcto si es **uno de los tres valores constantes** VALOR1, VALOR2 o VALOR3.

$$(\text{num} = \text{VALOR1} \text{ o } \text{num} = \text{VALOR2} \text{ o } \text{num} = \text{VALOR3})$$

Continuación ejemplo 6

c) El número es correcto si se **encuentra en el rango** definido por dos valores constantes MINIMO y MAXIMO, o bien es igual al valor constante VALOR1

$(\text{MINIMO} \leq \text{num} \leq \text{MAXIMO} \text{ o } \text{num} = \text{VALOR1})$

d) El número es correcto si se **encuentra en el rango** definido por dos valores constantes MINIMO1 y MAXIMO1, **o bien** en el definido por los valores constantes MINIMO2 y MAXIMO2.

$(\text{MINIMO1} \leq \text{num} \leq \text{MAXIMO1}) \text{ o }$

$(\text{MINIMO2} \leq \text{num} \leq \text{MAXIMO2})$

Análisis ejemplo 6

Análisis E-P-S

Entradas: numero entero ($\text{num} \in \mathbb{Z}$)

Proceso:

- **Si** ($\text{MINIMO} \leq \text{num} \leq \text{MAXIMO}$) o
($\text{num} = \text{VALOR1}$ o $\text{num} = \text{VALOR2}$ o $\text{num} = \text{VALOR3}$) o
($\text{MINIMO} \leq \text{num} \leq \text{MAXIMO}$ o $\text{num} = \text{VALOR1}$) o
($\text{MINIMO1} \leq \text{num} \leq \text{MAXIMO1}$ o
 $\text{MINIMO2} \leq \text{num} \leq \text{MAXIMO2}$) **el número es correcto**

si el número no cumple ninguna de estas condiciones es incorrecto

Salidas: Mensaje indicando si el número es o no correcto

Diseño ejemplo 6

numeroCorrecto		
	{pre: ninguna}	{pos: ninguna }
1	Escribir “Dame un numero entero”	-num: Entero. Numero a verificar su correctitud.
2	Leer num	
3	Si $((4 \leq \text{num} \wedge \text{num} \leq 65) \vee (\text{num} = 3 \vee \text{num} = 12 \vee \text{num} = 23) \vee (4 \leq \text{num} \wedge \text{num} \leq 65 \vee \text{num} = 3) \vee (34 \leq \text{num} \wedge \text{num} \leq 128 \vee 43 \leq \text{num} \wedge \text{num} \leq 45))$ entonces Escribir “Correcto” sino Escribir “ERROR - numero incorrecto” fsi	
1	num = 20 -> Correcto	Caso exitoso
2	num = -> ERROR - numero incorrecto	Caso exitoso

Codificación ejemplo 6

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

```
#define MINIMO 4  
#define MAXIMO 65  
#define VALOR1 3  
#define VALOR2 12  
#define VALOR3 23  
#define MINIMO1 34  
#define MAXIMO1 128  
#define MINIMO2 43  
#define MAXIMO2 45
```

Utilizando
constantes
globales

Codificación ejemplo 6

```
void main ()
{
    int num;

    print (“Introduzca el numero\n”);
    scanf (“%d”, &num);
    if ((num >= MINIMO && num <= MAXIMO) ||
        (num == VALOR1 || num == VALOR2 || num == VALOR3) ||
        (num >= MINIMO && num <= MAXIMO || num ==
        VALOR1) || (num >= MINIMO1 && num <= MAXIMO1 ||
        num >= MINIMO2 && num <= MAXIMO2))
        printf (“CORRECTO\n”);
    else
        printf (“ERROR - numero incorrecto\n”);
}
```

Ejercicio. Cambiar la codificación para que utilice constantes locales

Estructuras de decisión doble anidadas

Pseudocódigo en Español

```
Si (<condición1>) entonces
    sentencias1
sino
    Si (<condición2>) entonces
        sentencias2
    sino
        Si (<condición3>) entonces
            sentencias3
        fsi
    fsi
fsi
```

Codificación en C

```
if (<condición1>)
{
    sentencias1
}
else if(<condición2>)
{
    sentencias2
}
else if(<condición3>)
{
    sentencias3
}
```

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

```
void main ()
```

```
{
```

```
    int a1;
```

```
    printf (“Introduzca un valor entero\n”);
```

```
    scanf (“%d”, &a1);
```

```
    if (a1 >= 0 )
```

```
        if (a1 <= 10 )
```

```
            printf (“el numero esta entre 0 y 10\n”);
```

```
        else
```

```
            printf (“el numero no esta entre 0 y 10\n”);
```

```
}
```

Ejemplo 8

```
#include <stdio.h>

void main ()
{
    int a1;
    printf (“Introduzca un valor entero\n”);
    scanf ("%d", &a1);
    if (a1 >= 0 )
    {
        if ( a1 <= 10 )    printf (“el numero esta entre 0 y 10\n”);
        a1++;
    }
    else    printf (“el numero es negativo\n”);
}
```

Ejemplo 9

Dados tres números enteros, decidir cuál es el central

Análisis E-P-S

Entradas: $A, B, C \in \mathbb{Z}$

Proceso: - Determinar si A es central o B es central o C es central

$A > B > C \Rightarrow B$ es central

$A > C > B \Rightarrow C$ es central

$B > A > C \Rightarrow A$ es central

$B > C > A \Rightarrow C$ es central

$C > A > B \Rightarrow A$ es central

$C > B > A \Rightarrow B$ es central

Salida: valor central ($\text{central} \in \mathbb{Z}$)



Diseño ejemplo 9

numeroCentral		
{pre: ninguna}		{pos: central $\in \mathbb{Z}$ }
1 2 3	<pre>Escribir (“Introduzca tres números enteros”) Leer (A, B, C) Si (A > B) entonces Si (B > C) entonces central = B sino // B <= C Si (A > C) entonces central = C sino // A <= C central = A fsi fsi sino // A <= B Si (A > C) entonces central = A sino // A <= C Si (C > B) entonces central = B sino // C <= B central = C fsi fsi fsi Escribir (“el número central es = “, central)</pre>	<p>-A, B, C: <i>Entero</i>. Números leídos para encontrar cual de ellos es el central</p> <p>-central: <i>Entero</i>. Número central</p>
1 2	<p>A = 20, B = 40, C = 50 -> central = 40</p> <p>A = 35, B = 10, C = 100 -> central = 35</p>	<p>Caso exitoso</p> <p>Caso exitoso</p>

Codificación ejemplo 9

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

```
void main ()
```

```
{
```

```
    int A, B, C, central;
```

```
    printf ("Introduzca tres números  
    enteros\n");
```

```
    scanf ("%d%d%d", &A, &B, &C);
```

```
    if (A > B)
```

```
        if (B > C)
```

```
            central = B;
```

```
        else // B <= C
```

```
            if (A > C)
```

```
                central = C;
```

```
            else // A <= C
```

```
                central = A;
```

```
    else // A <= B
```

```
        if (A > C)
```

```
            central = A;
```

```
        else // A <= C
```

```
            if (C > B)
```

```
                central = B;
```

```
            else // C <= B
```

```
                central = C;
```

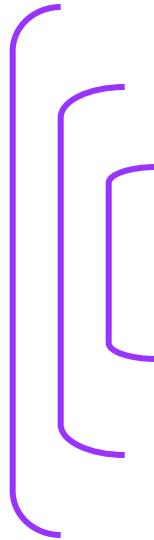
```
    printf ("el número central es =  
    %d\n", central);
```

```
}
```


Ejemplos de anidamiento válido



(a)



(b)



(c)



(d)



(e)

Realice el análisis E-P-S, algoritmo y codificación para cada uno de los siguientes problemas:

1. 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"
2. Leer el valor de **un año** e indicar si es o no un **año bisiesto**. Recordar la regla:
"Un año es bisiesto si es divisible por 400 o si es divisible por 4 pero no por 100"

3. Leer un instante de **tiempo** expresado en **horas** y **minutos** y escribir como respuesta uno de los mensajes "Buenos días", "Buenas tardes" o "Buenas noches", de acuerdo con las siguientes reglas:
- a) Se considera que es de día desde las 7:30 horas hasta las 14:00 horas.
 - b) Se considera que es por la tarde desde las 14:01 horas hasta las 20:30 horas.
 - c) Se considera que es de noche desde las 20:31 horas hasta las 7:29 horas.

4. Leer un caracter y deducir si está o no comprendido entre las letras I y M ambas inclusive.
5. Emitir la factura correspondiente a una compra de un artículo determinado del que se adquieren una o varias unidades. El IVA a aplicar es del 12% y si el precio bruto (precio de venta + IVA) es mayor de 50.000 Bs., se aplicará un descuento del 5%.
6. Leer las coordenadas cartesianas (x, y) de un punto del plano y calcular e imprimir el cuadrante al cual pertenece el punto (I, II, III, IV).

7. Se desea obtener la **nómina semanal -salario neto-** de los empleados de una empresa cuyo trabajo se **paga por horas** y del modo siguiente:
- Las horas inferiores o iguales a 35 horas (normales) se pagan a una tarifa determinada que se debe introducir por teclado al igual que el número de horas.
 - Las horas superiores a 35 se pagarán como extras al precio de 1.5 horas normales.
 - Los impuestos a deducir a los trabajadores varían en función de su sueldo mensual:
 - Sueldo \leq 80.000 libre de impuestos
 - Los siguientes 100.000 Bs. al 20%
 - El resto al 30%

8. Resolver una ecuación de primer grado. Si la ecuación es $ax + b = 0$, las posibles soluciones son:
 8. $a \neq 0$ entonces $x = -b/a$
 9. $a = 0$ y $b \neq 0$ entonces la solución es imposible
 10. $a = 0$ y $b = 0$ entonces la solución es indeterminada
9. Escribir un programa que convierta de grados Fahrenheit a Celsius y viceversa. El programa deberá leer en primer lugar un número entero que indicará si se va a convertir de grados Fahrenheit a Celsius (1) o de grados Celsius a Fahrenheit (2). Seguidamente leerá el valor de la temperatura, realizará la conversión aplicando la fórmula adecuada e imprimirá.