



Unidad 2. La lógica de programación

Tema 4. Estructuras de decisión

Estructuras de decisión simple y doble

▶ Contenido:

- ▶ Conceptos básicos
- ▶ Tipos:
 - ▶ Simple
 - ▶ Doble
 - ▶ Múltiple
 - varias condiciones
 - una única condición
- ▶ Representación algorítmica
- ▶ Codificación

▶ Objetivo:

Desarrollar habilidades en el uso de las estructuras de decisión

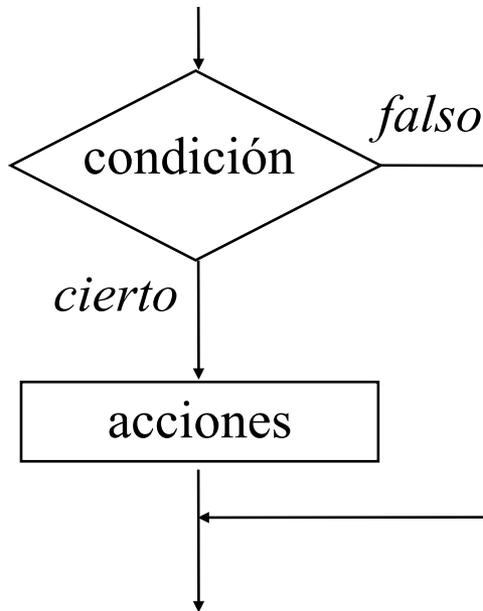
▶ Bibliografía:

- ▶ Deitel y Deitel, cap. 3.
- ▶ <http://www.ing.ula.ve/~ibc/pr1> o <http://www.webdelprofesor.ula.ve/ingenieria/ibc>
- ▶ Navas y Besembel, tema IV.
- ▶ Joyanes, sec.4.5, 4.7.

- ▶ Habilidad de definir el orden en el cual las sentencias de un programa son ejecutadas
 - ▶ 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

Estructuras de decisión simple

Diagrama de flujo



Pseudocódigo y TDSO

Si (<condicion>) entonces

S_1

....

S_n

fsi

Código en C/C++

```
if(<condicion>)
```

```
{  $S_1$ 
```

```
...
```

```
 $S_n$ 
```

```
}
```

Estructuras de decisión simple

Diseño	Código en C/C++
Si ($x > 0.0$) entonces Desplegar “x es positivo” fsi	<code>if(x > 0.0) cout << “x es positivo” << endl;</code>
Si ($a < b$) entonces $b = a + b$ fsi	<code>if(a < b) b=a+b;</code>

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

Estructuras de decisión simple

Diseño	Código en C/C++
Si (nota > 15) entonces conta=conta+1 Desplegar "Eximido" fsi	<pre>if(nota > 15) { conta++; cout << "Eximido" << endl; };</pre>
Si (x ≥ 0.0) entonces y = x + y cont=cont+1 fsi	<pre>if(x ≥ 0.0) { y += x; cont++; };</pre>

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

Ejemplo 1

```
#include <iostream.h>
using namespace std;
#define falso 0
#define cierto 1

int main ()
{
    int miPrimeraLogica, miSegundaLogica;
    miPrimeraLogica = cierto;
    miSegundaLogica = !miPrimeraLogica;
    if ( miSegundaLogica == cierto)
        cout << "miSegundaLogica debe tener el valor cierto"
<< endl;
    if ( !miSegundaLogica == cierto)
        cout << "miSegundaLogica debe tener el valor falso"
<< endl;
    return 0;
}
```

miPrimeraLogica=
miSegundaLogica=

¿Qué valores tienen las variables al finalizar el programa?

Ejercicio resuelto 1

▶ Enunciado del problema:

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 en E-P-S

Entrada	Variable	Descripción	Tipo de dato	Rango válido
	X Y Z	Primer número Segundo número Tercer número	Entero	El rango de un entero en la computadora
Proceso	Si $X+Y=Z$ o $X+Z=Y$ o $Y+Z=X$ hay iguales Si $X+Y \neq Z$ y $X+Z \neq Y$ y $Y+Z \neq X$ son diferentes			
Salida	Ninguna	Desplegar el mensaje “Iguales” si se da la condición, de lo contrario desplegar el mensaje “Distintos”	No Aplica (NA)	NA

Ejercicio resuelto 1. Diseño

Sep. 2010

iguales O distintos

{pre: }

{pos: }

<p>1 2 3</p>	<p>X, Y, Z = valor suministrado Si $(X + Y = Z \vee X + Z = Y \vee Y + Z = X)$ entonces Desplegar "Iguales" fsi Si $(X + Y \neq Z \wedge X + Z \neq Y \wedge Y + Z \neq X)$ entonces Desplegar "Distintos" fsi</p>	<p>X, Y, Z: Entero. Valores a comparar para determinar si la suma de cualesquiera dos de ellos es igual al tercero</p>
<p>1 2 3</p>	<p>X=1, Y=1, Z=1 => Distintos X=2, Y=3, Z=6 => Distintos X=0, Y=0, Z=0 => Iguales</p>	<p>Caso exitoso Caso exitoso Caso exitoso</p>

Ejercicio resuelto 1. Codificación en C++

```
#include <iostream>

using namespace std;

int main ()
{   int X, Y, Z;

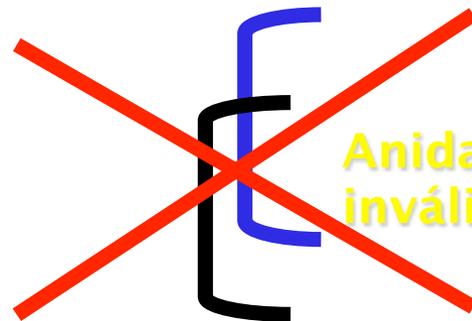
    cout << "Introduzca tres números enteros" << endl;
    cin >> X >> Y >> Z;
    if ((X + Y == Z) || (X + Z == Y) || (Y + Z == X))
        cout << "Iguales" << endl;
    if ((X + Y != Z) && (X + Z != Y) && (Y + Z != X))
        cout << "Distintos" << endl;
    return 0;
}
```

Anidamiento de las estructuras de decisión

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



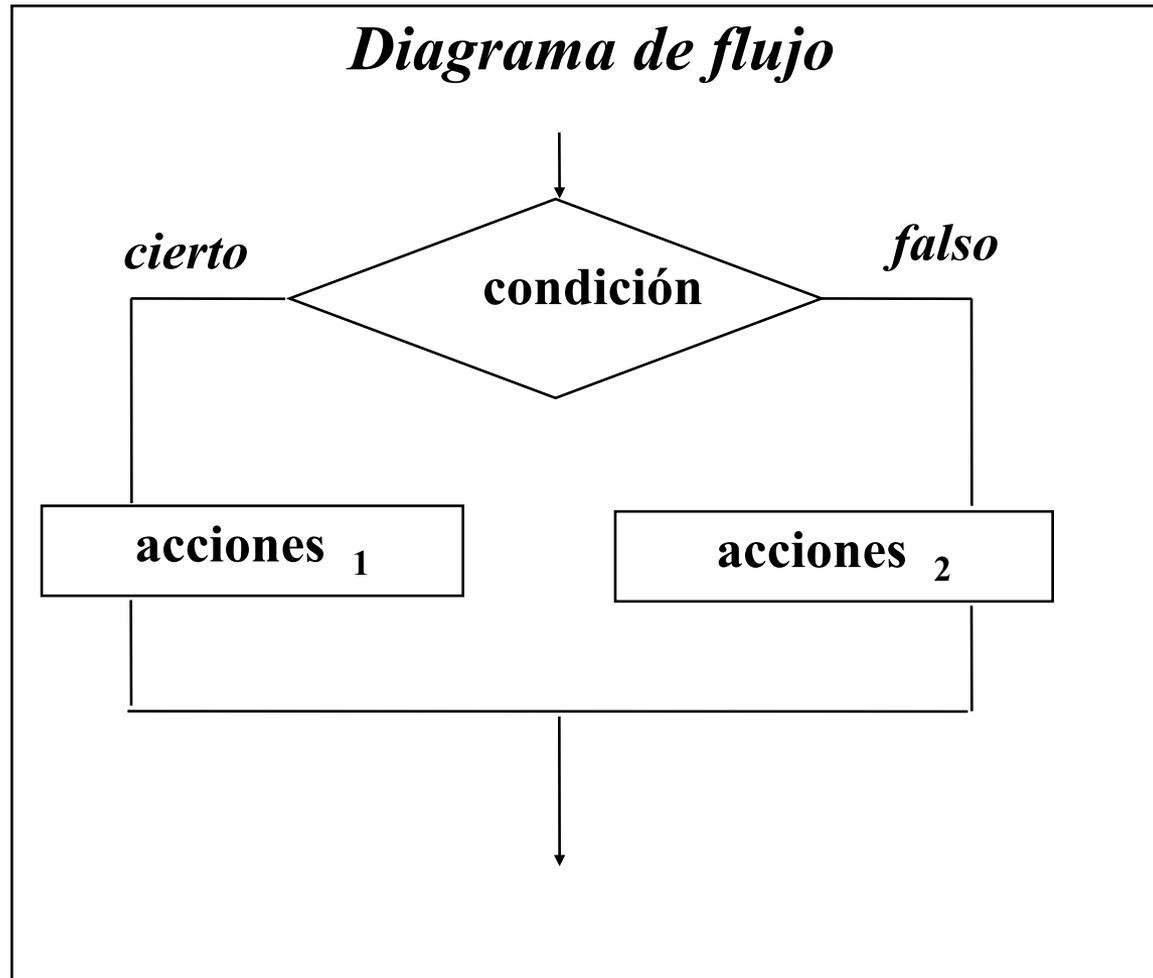
Anidamiento válido



Anidamiento inválido



Estructuras de decisión doble



Estructuras de decisión doble

Pseudocódigo y TDSO

Si(<condición>) entonces

S_{11}

...

S_{1n}

sino

S_{21}

...

S_{2m}

fsi

Código en C/C++

```
if(<condición>)
```

```
{  
     $S_{11}$ 
```

```
    ...
```

```
     $S_{1n}$ 
```

```
}
```

```
else
```

```
{  
     $S_{21}$ 
```

```
    ...
```

```
     $S_{2m}$ 
```

```
}
```

Estructuras 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** determinado grupo de sentencias S_{2k} , $k = 1, \dots, m$

Estructuras de decisión doble

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

```
#include <iostream>
using namespace std;
int main()
{
    int miEntero;
    cout << "Introduzca un número entero" << endl;
    cin >> miEntero;
    if ( miEntero == 0 )
        cout << "miEntero es igual a cero" << endl;
    else
        cout << "miEntero es diferente de cero" << endl;
    cout << "\n terminó el programa" << endl;
    return 0;
}
```

Ejercicio resuelto 2

► Enunciado del problema:

Calcular el salario mensual de un empleado, sabiendo que éste se calcula según las horas semanales trabajadas y el precio especificado por hora. Si trabaja mas de 40 horas semanales, entonces las horas extras se pagan a 1,5 veces la hora ordinaria.

► Análisis E-P-S

Entrada	Variable	Descripción	Tipo de dato	Rango válido
	HST	Horas semanales trabajadas	Real	≥ 0
	PH	Precio por hora	Real	≥ 0
Proceso	Si $HST \leq 40 \Rightarrow$ calcular el salario mensual (SM) como $HST * PH$ Si $HST > 40 \Rightarrow$ calcular el salario mensual como $40 * PH + PH * 1,5 *$			
Salida	SM	Salario mensual	Real	≥ 0

Ejercicio resuelto 2. Diseño

Sep. 2010

SalarioMensual

{pre:}

{pos: SM ≥ 0}

1	Desplegar “Introduzca el número de horas trabajadas”	
2	HST = valor suministrado	
3	Desplegar “Introduzca el precio por hora “	
4	PH = valor suministrado	
5	Si ($HST \leq 40$) entonces SM=HST*PH sino SM = 40 * PH + PH*1,5*(HST-40)	-HST: Real+. Horas semanales trabajadas. -PH: Real+. Precio por horas. -SM: Real+. Sueldo mensual.
6	fsi Desplegar “Sueldo mensual = “, SM	
1	HST = 44.0, PH = 850.0 -> SM = 39100.0	Caso exitoso
2	HST = 0.0, PH = 950.0 -> SM = 0.0	Caso borde
3	HST = 47.0, PH = 0.0 -> SM = 0.0	Caso borde
4	HST = 0.0, PH = 0.0 -> SM = 0.0	Caso borde

Estructuras de decisión simple y dobles anidadas

Las estructuras de decisión simples o dobles se introducen unas en el interior de otras

```
Si (<condición1>) entonces
  sentencias
sino
  Si (<condición2>) entonces
    sentencias
  sino
    Si (<condición 3>) entonces
      sentencias
    fin_si
  fin_si
fin_si
```

```
if (<condición1>)
else.....
    if (<condición2>)
        .....
    else
        if (<condición3>)
            .....
```

```
#include <stdio.h>
int main ()
{
    int a1;
    printf("Introduzca un valor entero");
    scanf("%d", &a1);
    if ( a1 >= 0 )
        if ( a1 <= 10 )
            printf("El número está entre 0 y 10\n");
        else
            printf("El número no está entre 0 y 10\n");
    return 0;
}
```

```
#include <stdio.h>
int main ()
{
    int a1;
    printf("Introduzca un valor entero\n");
    scanf("%d", &a1);
    if ( a1 >= 0 )
    {
        if ( a1 <= 10 )
            printf("El número está entre 0 y 10\n");
        a1++;
    }
    else
        printf("El número es negativo\n");
    return 0;
}
```

Ejercicio resuelto 3

► Enunciado del problema:

Escriba un programa que pida el resultado obtenido al lanzar un dado de seis caras y muestre por pantalla el número en letras de la cara opuesta al resultado obtenido. La caras opuestas de un dado son: 1-6, 2-5 y 3-4.

► Análisis E-P-S

Entrada	Variable	Descripción	Tipo de dato	Rango válido
	numero	Número de la cara del dado leído de teclado	Entero	Entre 1 y 6
Proceso	Si numero es 1 ⇒ desplegar que en la cara opuesta está el seis Si numero es 2 ⇒ desplegar que en la cara opuesta está el cinco Si numero es 3 ⇒ desplegar que en la cara opuesta está el cuatro Si numero es 4 ⇒ desplegar que en la cara opuesta está el tres Si numero es 5 ⇒ desplegar que en la cara opuesta está el dos Si numero es 6 ⇒ desplegar que en la cara opuesta está el uno Cualquier otro valor para numero es incorrecto			
Salida	ninguna	No Aplica (NA)	NA	NA

Ejercicio resuelto 3: Diseño

carasDelDado		{pos: }
{pre: }		
<p>1 numero = valor suministrado</p> <p>2 Si (numero = 1) entonces Desplegar “En la cara opuesta está el seis” sino Si (numero = 2) entonces Desplegar “En la cara opuesta está el cinco” sino Si (numero = 3) entonces Desplegar “En la cara opuesta está el cuatro” sino Si (numero = 4) entonces Desplegar “En la cara opuesta está el tres” sino Si (numero = 5) entonces Desplegar “En la cara opuesta está el dos”</p>		<p>-numero: Entero. Valor numérico de la cara del dado. numero ∈ {1, ..., 6}</p>

Ejercicio resuelto 3: Diseño (cont)

	<pre> sino Si (numero = 6) entonces Desplegar "En la cara opuesta está el uno" sino Desplegar "Error ... Número incorrecto" fsi fsi fsi fsi fsi fsi fsi </pre>	
<ol style="list-style-type: none"> 1 2 3 4 5 6 7 	<pre> numero = 1 => En la cara opuesta está el seis numero = 2 => En la cara opuesta está el cinco numero = 3 => En la cara opuesta está el cuatro numero = 4 => En la cara opuesta está el tres numero = 5 => En la cara opuesta está el dos numero = 6 => En la cara opuesta está el uno numero = -245 => Error ... Número incorrecto </pre>	<pre> Caso exitoso </pre>

Estructura de decisión múltiple con varias condiciones

Si (condicion_1) entonces

.....

sino Si (condicion_2) entonces

.....

sino Si (condicion_3) entonces

.....

.....

sino Si (condicion_k) entonces

.....

sino

.....

fsi

```
if (condicion_1)
```

```
{
```

```
.....
```

```
}
```

```
else if (condicion_2)
```

```
{
```

```
.....
```

```
}
```

```
else if (condicion_3)
```

```
{
```

```
.....
```

```
.....
```

```
}
```

```
else if (condicion_k)
```

```
{
```

```
.....
```

```
}
```

```
else
```

```
{
```

```
.....
```

```
}
```

Ejercicio resuelto 3: decisión múltiple con varias condiciones

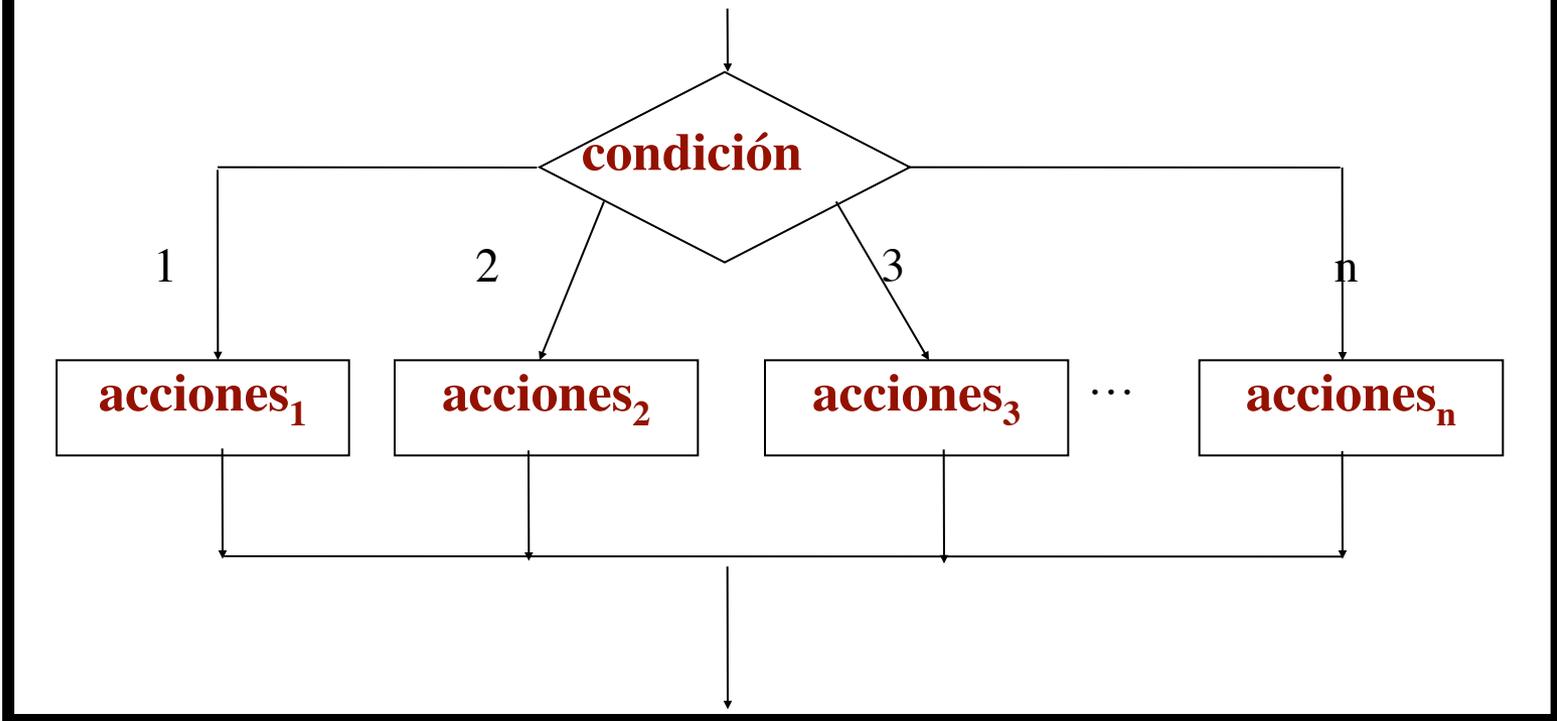
carasDelDadoDM		{pos: }
{pre: }		
1 2	<pre> numero = valor suministrado Si (numero = 1) entonces Desplegar "En la cara opuesta está el seis" sino Si (numero = 2) entonces Desplegar "En la cara opuesta está el cinco" sino Si (numero = 3) entonces Desplegar "En la cara opuesta está el cuatro" sino Si (numero = 4) entonces Desplegar "En la cara opuesta está el tres" sino Si (numero = 5) entonces Desplegar "En la cara opuesta está el dos" sino Si (numero = 6) entonces Desplegar "En la cara opuesta está el uno" sino Desplegar "Error..... Número incorrecto" fsi </pre>	<pre> -numero: Entero. Valor numérico de la cara del dado. numero ∈ {1, ..., 6} </pre>

Codificación del ejercicio resuelto 3: decisión múltiple con varias condiciones

```
#include <iostream>
using namespace std;
int main()
{
    int numero;
    cout << "Introduzca el número del dado " << endl;
    cin >> numero;
    if ( numero == 1 )
        cout << "En la cara opuesta está el seis (6)" << endl;
    else if (numero == 2);
        cout << "En la cara opuesta está el cinco (5)" << endl;
    else if (numero == 3);
        cout << "En la cara opuesta está el cuatro (4)" << endl;
    else if (numero == 4)
        cout << "En la cara opuesta está el tres (3)" << endl;
    else if (numero == 5);
        cout << "En la cara opuesta está el dos (2)" << endl;
    else if (numero == 6);
        cout << "En la cara opuesta está el uno (1)" << endl;
    else
        cout << "Error.... Número incorrecto" << endl;
    return 0;
}
```

Estructuras de decisión múltiple con una única condición

Diagrama de flujo



Estructuras de decisión múltiple con una única condición

```
En caso_de <expresión> hacer
  <lista1>: <sentencias1>
  <lista2>: <sentencias2>
  .....
  [sino: <sentenciasN>]
```

FinCaso

TDSO

```
En caso que  $X_i = E_i$ :  $Y_1, \dots, Y_n = E'_1, \dots, E'_n$ 
  ....
   $E_j$ :  $S_1, \dots, S_m = F_1, \dots, F_m$ 
  en otro caso
     $Z_1, \dots, Z_p = G_1, \dots, G_p$ 
```

```
switch (<expresión>)
{
  case <valor1>:
    <sentencias1>
    break;
  case <valor2>:
    <sentencias2>
    break;
  .....
  default: // Opcional
    <sentenciasN>
}
```

Ejercicio resuelto 1. Diseño con decisión múltiple de una condición

carasDelDadoDMcaso		
	{pre: }	{pos: }
1	numero = valor suministrado	-numero: Entero. Valor numérico de la cara del dado. $\text{numero} \in \{1, \dots, 6\}$
2	En caso que numero= 1: Desplegar “En la cara opuesta está el seis” 2: Desplegar “En la cara opuesta está el cinco” 3: Desplegar “En la cara opuesta está el cuatro” 4: Desplegar “En la cara opuesta está el tres” 5: Desplegar “En la cara opuesta está el dos” 6: Desplegar “En la cara opuesta está el uno” en otro caso Desplegar “Error..... Número incorrecto”	
1	numero=1 => En la cara opuesta está el seis	Caso exitoso
2	numero=2 => En la cara opuesta está el cinco	Caso exitoso
3	numero=3 => En la cara opuesta está el cuatro	Caso exitoso
4	numero=4 => En la cara opuesta está el tres	Caso exitoso
5	numero=5 => En la cara opuesta está el dos	Caso exitoso
6	numero=6 => En la cara opuesta está el uno	Caso exitoso
7	numero=456 => Error..... Número incorrecto	Caso exitoso

```
#include <iostream>
using namespace std;
int main()
{
    int numero;
    cout << "Introduzca el número del dado " << endl;
    cin >> numero;
    switch (numero)
    {
        case 1: cout << "En la cara opuesta está el seis (6)" << endl;
                break;
        case 2: cout << "En la cara opuesta está el cinco (5)" << endl;
                break;
        case 3: cout << "En la cara opuesta está el cuatro (4)" << endl;
                break;
        case 4: cout << "En la cara opuesta está el tres (3)" << endl;
                break;
        case 5: cout << "En la cara opuesta está el dos (2)" << endl;
                break;
        case 6: cout << "En la cara opuesta está el uno (1)" << endl;
                break;
        default: cout << "Error.... Número incorrecto" << endl;
    }
    return 0;
}
```

Decisión simple y doble anidadas

Si(condición) entonces

$X_1, \dots, X_n = E_1, \dots, E_n$

sino

$Y_1, \dots, Y_m = E'_1, \dots, E'_m$

fsi

```
If(condicion1)
```

```
{ .....
}
```

```
else
```

```
  if(condicionk)
```

```
  { .....
  }
```

Decisión múltiple con varias condiciones

Si(condición) entonces

$X_1, \dots, X_n = E_1, \dots, E_n$

sino Si(condición1) entonces

$Y_1, \dots, Y_m = E'_1, \dots, E'_m$

sino Si(condición2) entonces

.....

fsi

```
if (condicion1)
```

```
{ .....
}
```

```
.....
```

```
else if (condicion_k)
```

```
{ .....
}
```

```
else
```

```
{ .....
}
```

Decisión múltiple con una única condición

En caso que $X_i = E_i$: $X'_1, \dots, X'_m = E'_1, \dots, E'_m$

.....

$= E_j$: $Y_1, \dots, Y_p = F_1, \dots, F_p$

en otro caso

$Z_1, \dots, Z_q = G_1, \dots, G_q$

```
switch (expresión)
```

```
{ case valor1: { ..... }
                    break;
```

```
.....
```

```
  default: // Opcional
```

```
  { ..... }
```

```
}
```

- ▶ ¿Qué es flujo de control?
- ▶ ¿Qué son las estructuras de decisión?
- ▶ ¿Cuáles son los tipos de estructuras de decisión?
- ▶ ¿Cómo son las estructuras de decisión simple?
- ▶ ¿Cómo son las estructuras de decisión doble?
- ▶ ¿Cómo se anidan las estructuras de decisión?
- ▶ ¿Cómo anidar la decisiones simples y dobles?
- ▶ ¿Qué es una estructura de decisión múltiple?
- ▶ ¿Cómo son las estructuras de decisión múltiple con varias condiciones?
- ▶ ¿Cómo es la estructura de decisión múltiple con una única condición?

Resumen
¿Cuáles son los conceptos relevantes de esta clase?



1. Si $x = 15$ determinar su valor después de la ejecución del siguiente segmento de programa en C++:
$$\text{if}(x > 0) \quad x += 32;$$

2. Si $z = 6.7$ determinar el valor de la variable `conta` después de la ejecución del siguiente segmento de programa en C++:
$$\begin{aligned} \text{conta} &= 0; \\ \text{neg} &= 0; \\ \text{if}(z < 0.0) \quad \text{neg} &= \text{neg} + z; \\ \text{conta} &= \text{neg} + 1; \end{aligned}$$



3. Leer un carácter y deducir si está o no comprendido entre las letras I y M ambas inclusive
4. 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%
5. Dado un reloj de 24 horas y un tiempo representado en forma de entero de la forma hhmm, donde hh representa las horas (00 a 23) y mm representa los minutos (00 a 59), determinar el tiempo después de que han pasado h horas y m minutos



Ejercicios

6. 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
7. Determinar el precio de un pasaje de ida y vuelta en avión, conociendo la distancia a viajar y sabiendo que si el número de días de estancia es superior a s y la distancia superior a d km. el pasaje tiene una reducción del $p\%$. El precio por km. es b Bs.

8. 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"
9. 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%

10. 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 < 80.000 libre de impuestos

Los siguientes 100.000 Bs. al 20%

El resto al 30%

1. Dada una fecha con el formato dd, mm, aaaa, escriba un programa para validar si el día es válido con respecto al mes
2. Escriba un programa en C que a partir de una nota expresada en forma numérica imprima su correspondiente literal, según la siguiente correspondencia: 0 a 9 REPROBADO, 10 y 14 APROBADO, 15 y 18 NOTABLE, 19 y 20 SOBRESALIENTE. Para cualquier otro valor de la nota numérica se imprimirá el mensaje VALOR INCORRECTO
3. Escriba un programa tal que, dado el valor de un día de la semana entre 1 y 7, escriba si es un día es laboral o es fin de semana

12. En la siguiente tabla se muestra el número de camas de las habitaciones de una casa rural, además de la planta donde está ubicada cada una de ellas:

Muestre un menú con el listado de las habitaciones de la casa rural

Pida por teclado el número (dato entero) asociado a una habitación

Muestre por pantalla la planta y el número de camas de la habitación seleccionada

Nota: Si el número introducido por el usuario, no está

Habitación	Camas	Planta	Habitación	Camas	Planta
1. Azul	2	Baja	4. Rosa	2	Primera
2. Roja	1	Baja	5. Lila	1	Segunda
3. Verde	3	Primera	6. Naranja	3	Segunda