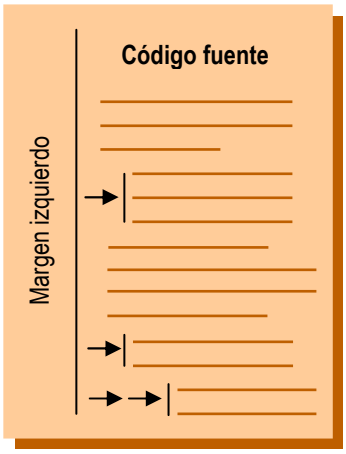
PRÁCTICA 3: ESTRUCTURAS DE CONTROL CONDICIONALES

OBJETIVOS:

- ✓ Revisar el concepto de algoritmo y entender la necesidad del diseño de algoritmos en el estudio y resolución de programas
- ✓ Comprender y diferenciar perfectamente los distintos tipos de instrucciones utilizados en el diseño de algoritmos
- ✓ Conocer y manejar con habilidad las sentencias de control condicionales

*No olvides que antes de pasar a la fase de diseño deberemos **analizar** exhaustivamente **el problema***

Recuerda también que la escritura de un programa exige la "**indentación**" o "**sangría**" del texto en el margen izquierdo de las diferentes líneas, lo que facilita el entendimiento y comprensión del programa realizado. Cualquier sentencia dentro del cuerpo de otra sentencia, debe sangrarse.



Ejercicio Resuelto 1. **Implementa un algoritmo en C que tenga como entrada el precio de venta de un piso, y que produzca como salida la comisión del vendedor, teniendo en cuenta que si el precio de venta es inferior a 100.000 €, la comisión es del 4%, si el precio está entre 100.000 € y 300.000 € la comisión es del 3% y si el precio es superior a 300.000 € la comisión es del 2'5 %.**

Solución a)

```
#include <iostream>
using namespace std;
int main()
{
    long int precio;
    float comision;

    cout << "\nIntroduzca el precio del piso:";
    cin >> precio;

    if  (precio < 100000)  {
         comision = precio * 0.04;
    }
    if  (precio >= 100000 && precio <= 300000)   {
         comision = precio * 0.03;
    }
    if  (precio > 300000)  {
         comision = precio * 0.025;
    }
    cout << "\nLa comision es de " << comision << "euros.";

}
```

En este algoritmo hay tres condiciones (la segunda de ellas compuesta) que se ejecutan secuencialmente. Independientemente de que este algoritmo produzca el resultado deseado, es mejorable en cuanto a su eficiencia, ya que tal y como está diseñado, las tres condiciones serán evaluadas siempre que ejecutemos el programa, ralentizando por tanto su ejecución.

Solución b)

```
#include <iostream>
using namespace std;
int main()
{
    long int precio;
    float comision;

    cout << "\nIntroduzca el precio del piso:";
    cin >> precio;

    if (precio < 100000)  {
        comision = precio * 0.04;
    }
    else  {
         if (precio < 300000)   {
              comision = precio * 0.03;
         }
         else  {
              comision = precio * 0.025;
         }
   }
    cout << "\nLa comision es de " << comision << "euros.";

}
```

En esta solución las tres condiciones son simples, y además, gracias a la estructura utilizada ("if-else" anidados), hemos conseguido una mayor eficiencia, ya que en el caso de que el precio sea inferior a 100000, sólo se evaluará una condición (la primera), ejecutándose a continuación la sentencia de salida. En cualquier caso, como máximo se evaluarán dos condiciones, ya que tanto si el precio es inferior o superior a 300000, sólo se evaluarán dos condiciones.

Solución c) Esta solución es idéntica a la anterior, y simplemente se propone para mostrar que los delimitadores de bloques solamente son necesarios cuando dentro del cuerpo del bloque "if" o del bloque

"else" hay más de una instrucción o de una estructura (tened en cuenta que a estos efectos, una estructura completa "if-else" es considerada como una sola instrucción). El algoritmo por tanto podría quedar así:

```
#include <iostream>
using namespace std;
int main()
{
    long int precio;
    float comision;

    cout << "\nIntroduzca el precio del piso:";
    cin >> precio;

    if (precio < 100000)
         comision = precio * 0.04;
    else
         if (precio < 300000)
             comision = precio * 0.03;
         else
             comision = precio * 0.025;
    cout << "\nLa comision es de " << comision << "euros.";

}
```

Solución d) Supongamos ahora que nos dicen que la mayoría de los precios de las viviendas son superiores a 300000 euros. Evidentemente esto no afecta a los cálculos que debe realizar el programa y por tanto seguirían siendo válidas las soluciones anteriores. Pero, ¿podríamos hacer algo por mejorar la eficiencia? Sí, y lo conseguiríamos cambiando el orden de las condiciones.

```
 #include <iostream>
using namespace std;
int main()
{
    long int precio;
    float comision;

    cout << "\nIntroduzca el precio del piso:";
    cin >> precio;

    if (precio >= 300000)
         comision = precio * 0.025;
    else
         if (precio >= 100000)
             comision = precio * 0.03;
         else
             comision = precio * 0.04;
    cout << "\nLa comision es de " << comision << "euros.";

}
```

De esta forma, en la mayoría de las ocasiones (cuando el precio sea superior a 300000) sólo se evaluará la primera condición. Observad cómo se han implementado las condiciones contrarias a las de la solución anterior.

Ejercicio Resuelto 2. **Implementa un algoritmo en C que realice las cuatro operaciones básicas de una calculadora (suma, resta, multiplicación y división). El programa debe leer los dos números y la operación y devolver el resultado.**

Solución a)

```
#include <iostream>

using namespace std;
int main()
{
     float a, b, resultado;
     char op;

     cout<<"\n Introduce el primer término: ";
     cin>> a;

     cout<<"\n Introduce el segundo término: ";
```

```
     cin>> b;

     cout<<"\n Introduce la operación";
     cin>>op;

     if (op == '+')
          resultado = a + b;
     else
          if (op == '-')
               resultado = a - b;
          else
               if (op == '*')
                    resultado = a * b;
               else
                    if (op == '/')
                         resultado = a / b;

 cout<<"El resultado es: "<<resultado<<"\n";

}
```

Solucion b)

```
#include <iostream>

using namespace std;
int main()
{
     float a, b, resultado;
     char operacion;

     cout<<"\n Introduce el primer término: ";
     cin>> a;

     cout<<"\n Introduce el segundo término: ";
     cin>> b;

     cout<<"\n Introduce la operación";
     cin>>op;

     switch (operacion){
          case '+': resultado = a+b;
                    break;
          case '-': resultado = a - b;
                    break;
          case '*': resultado = a * b;
                    break;
          case '/': resultado = a / b;
                    break;
     }

     cout<<"El resultado es: "<<resultado<<"\n";

}
```

En la solución b se ha utilizado la sentencia switch que permite escoger entre múltiples alternativas. Obtenemos así una solución más clara y eficiente que con la primera opción.

Ejercicio Resuelto 3. **Realiza un programa que calcule en índice de masa corporal de una persona. Se debe introducir el peso en kilogramos y la altura en metros, el índice de masa corporal se calcula con la siguiente fórmula: peso/(altura*altura). El programa debe devolvernos el tipo de peso que tenemos con respecto al IMC teniendo en cuenta la siguiente tabla.**

IMC	Tipo de peso
< 18.0	Inferior al normal
18.1 – 24.9	Normal
25.0 – 29.9	Sobrepeso
> 30.0	Obesidad

Ejemplo 1:

```
Introduce tu peso en Kg:
```

```
    75
    Introduce tu talla en m:
    1.80
    Tu IMC es: 23.1481. Normal
```

Solucion

```
#include <iostream>
using namespace std;
int main()
{
    float imc, altura, peso;


    cout << "Introduce tu peso ";
    cin >> peso;
    cout << "Introduce tu altura en metros ";
    cin >> altura;
    imc = peso/(altura*altura);

    if (imc<=18.0)
       cout<<"Tu IMC es:"<<imc<<" Peso inferior al normal"<<endl;
    else if (imc<25)
         cout<<"Tu IMC es:"<<imc<<"Peso normal"<<endl;
     else if (imc< 30)
           cout<<"Tu IMC es:"<<imc<<"Sobrepeso"<<endl;
          else cout<<"Tu IMC es:"<<imc<<"Obesidad"<<endl;
}
```