

Semestre I

Sesión 22

CARRERA PROFESIONAL

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**FUNDAMENTOS DE
PROGRAMACIÓN**

Tema

ESTRUCTURAS DE CONTROL ANIDADAS

Conceptos Basicos

Sentencias IF-ELSE Anidadas¹

Hasta este momento, las sentencias if implementan decisiones que implican una o dos alternativas. En esta sección se mostrará cómo se puede utilizar la sentencia if para implementar decisiones que implican diferentes alternativas.

Una sentencia if es anidada cuando la sentencia de la rama verdadera o la rama falsa es a su vez una sentencia if . Una sentencia if anidada se puede utilizar para implementar decisiones con varias alternativas o multi-alternativas.

Sintaxis

```
if (condición1)
    sentencia1
else if (condición2)
    sentencia2
.
.
.
else if (condiciónn)
    sentencian
else
    sentenciae
```

Ejemplo

```
// incrementar contadores de números positivos, números
// negativos o ceros

if (x > 0)
    num_pos = num_pos + 1;
else
```

¹ <https://combomix.net/wp-content/uploads/2017/03/Fundamentos-de-programación-4ta-Edición-Luis-Joyanes-Aguilar-2.pdf>

```
if (x < 0)
    num_neg = num_neg + 1;
else
    num_ceros = num_ceros + 1;
```

La sentencia if anidada tiene tres alternativas. Se incrementa una de las tres variables

(num_pos, num_neg y num_ceros) en 1, dependiendo de que x sea mayor que cero, menor que cero o igual a cero, respectivamente. Las cajas muestran la estructura lógica de la sentencia if anidada; la segunda sentencia if es la acción o tarea falsa (a continuación de else) de la primera sentencia if.

La ejecución de la sentencia if anidada se realiza como sigue: se comprueba la primera condición ($x > 0$); si es verdadera, num_pos se incrementa en 1 y se salta el resto de la sentencia if. Si la primera condición es falsa, se comprueba la segunda condición ($x < 0$); si es verdadera num_neg se incrementa en uno; en caso contrario se incrementa num_ceros en uno. Es importante considerar que la segunda condición se comprueba sólo si la primera condición es falsa.

Sangría en las sentencias if anidadas

El formato multibifurcación se compone de una serie de sentencias if anidadas, que se pueden escribir en cada línea una sentencia if. La sintaxis multibifurcación anidada es:

Formato 1

```
if (expresión_lógica1)
    sentencia1
else
    if (expresión_lógica2)
        sentencia2
    else
        if (expresión_lógica3)
            sentencia3
        else
            if (expresión_lógica4)
```

```

        sentencia4
else
    sentencia5

```

Formato 2

```
if (expresión_lógica1)
```

```
    sentencia1
```

```
else if (expresión_lógica2)
```

```
    sentencia2
```

```
else if (expresión_lógica3)
```

```
    sentencia3
```

```
else if (expresión_lógica4)
```

```
    sentencia4
```

```
else
```

```
    sentencia5
```

Ejemplos

1. if(x>0)

```
    if (y > 0)
```

```
        z = sqrt(x) + sqrt(y);
```

2. if(x>0)

```
    if (y > 0)
```

```
        z = sqrt(x) + sqrt(y);
```

```
    else
```

```
        cerr << "\n *** Imposible calcular z" << endl;
```

Comparación de sentencias if anidadas y secuencias de sentencias if

Los programadores tienen dos alternativas: (1) usar una secuencia de sentencias if; (2) una única sentencia if anidada. Por ejemplo, la sentencia if del Ejemplo se puede reescribir como la siguiente secuencia de sentencias if.

```

if (x > 0)
    num_pos = num_pos + 1;
if (x < 0)
    num_neg = num_neg + 1;
if (x == 0)
    num_ceros = num_ceros + 1;

```

Aunque la secuencia anterior es lógicamente equivalente a la original, no es tan legible ni eficiente. Al contrario que la sentencia if anidada, la secuencia no muestra claramente cuál es la sentencia a ejecutar para un valor determinado de x. Con respecto a la eficiencia, la sentencia if anidada se ejecuta más rápidamente cuando x es positivo ya que la primera condición ($x > 0$) es verdadera, lo que significa que la parte de la sentencia if a continuación del primer else se salta. En contraste, se comprueban siempre las tres condiciones en la secuencia de sentencias if. Si x es negativa, se comprueban dos condiciones en las sentencias if anidadas frente a las tres condiciones de las secuencias de sentencias if.

Una estructura típica if-else anidada permitida es:

```

if (número > 0)
{
    // ...
}
Else
{
    if (// ...)
    {
        // ...
    }
    else
    {
        if (// ...)
        {

```

```

        //
    }
} // ...
}

```

Algunos Ejemplos para poner en practica lo aprendido

1. Realizar un algoritmo que me permita realizar una calculadora basica, donde se pide al usuario ingresar dos numeros y un operador(+,-,*,/)

Algoritmo CalculadoraBasica

Escribir "Ingrese el primer número:"

Leer num1

Escribir "Ingrese el segundo número:"

Leer num2

Escribir "Ingrese el operador (+, -, *, /):"

Leer operador

Si operador = "+" Entonces

Escribir "Resultado:", num1 + num2

Sino Si operador = "-" Entonces

Escribir "Resultado:", num1 - num2

Sino Si operador = "*" Entonces

Escribir "Resultado:", num1 * num2

Sino Si operador = "/" Entonces

Si num2 ≠ 0 Entonces

Escribir "Resultado:", num1 / num2

Sino

Escribir "Error: No se puede dividir por cero."

FinSi

FinSi

FinAlgoritmo

2. Realizar un algoritmo que me permita verificar si un numero es Divisible por 2,3 o 5

Escribir "Ingrese un número:"

Leer numero

Si numero % 2 = 0 entonces

Escribir "El número es divisible por 2"

Sino si numero % 3 = 0 entonces

Escribir "El número es divisible por 3"

Sino si numero % 5 = 0 entonces

Escribir "El número es divisible por 5"

Sino

Escribir "El número no es divisible por 2, 3 ni 5"

Fin Si

