

Cátedra de COMPUTACIÓN

FACULTAD DE INGENIERÍA - UNA

La mejor manera de aprender Programación es
PROGRAMANDO !!!!!

CLASE NRO. 3

Estructuras de Control:

Durante la ejecución de un programa puede ser necesario **bifurcar, repetir o tomar decisiones**. C++ ofrece estructuras que permiten **controlar** estas situaciones.

Estructura Condicional: (sí y sí no)

if y else

Nomenclatura:

if (condición) {lista de instrucciones 1} **else** {lista de instrucciones 2}

Ejemplo: Sea **x** un numero cualquiera

```
if ( x > 0)
    {cout << x;
    cout << " es Positivo";}
else if ( x < 0)
    {cout << x;
    cout << " es Negativo";}
else cout << " es Negativo";
```

Estructura Repetitiva: (loops)

while

Nomenclatura:

while (condición) {lista de instrucciones}

- ◆ Repite la lista de instrucciones mientras se cumpla la condición.

Ejemplo:

```
# include (iostream.h)
```

```
int main()
```

```
{
```

```
int N; cout << "Introduzca un número para iniciar: "; cin >> N;cout << "\n";
```

```
while ( N > 0 )
```

```
    {cout << N << " , ";
```

```
    --N;}
```

```
cout << "\nFUEGO !!!!!\n";
```

```
return 0;
```

```
}
```

Introduzca un número para iniciar: **10**

10, 9, 8, 7, 6, 5, 4, 3, 2, 1,

FUEGO !!!!!

—

Estructura Repetitiva: (loops)

do - while

Nomenclatura:

do {lista de instrucciones} **while** (condición);

- ◆ Repite la lista de instrucciones mientras se cumpla la condición.

Ejemplo:

```
# include (iostream.h)
```

```
int main()
```

```
{
```

```
    unsigned long N;
```

```
    do {
```

```
        cout << "Introduzca un número ( 0 para finalizar): "; cin >> N;
```

```
        cout << "\nUd. Introdujo: " << N << "\n";
```

```
    }
```

```
while ( N != 0 );
```

```
return 0;
```

```
}
```

Introduzca un número (0 para finalizar): **1457**

Ud. Introdujo: 1457

Introduzca un número (0 para finalizar): **3289**

Ud. Introdujo: 3289

Introduzca un número (0 para finalizar): **0**

Ud. Introdujo: 0

—

Estructura Repetitiva: (loops)

for

Nomenclatura:

for (inicialización; condición; incremento){lista de instrucciones}

- ◆ Repite la lista de instrucciones, desde el valor inicial hasta la condición, aumentando el índice según el incremento.

Ejemplo 1:

```
# include (iostream.h)

int main() {
    for (int n=10; n>0; n--)
    {
        cout << n <<" ";
    }
    cout << "\nFUEGO\n";
    return 0;}

```

Ejemplo 2:

```
# include (iostream.h)
int main() { int n, i, veces;
    for (n = 0, i = 100; n != i; n++, i--)
    {
        cout << n <<" ", " << i <<"\n";
        veces++
    }
    cout << "\nSe repitió " <<veces <<"veces";
    return 0;}

```


10, 9, 8, 7, 6, 5, 4, 3, 2, 1,
FUEGO !!!!!

—

0, 100

1, 99

2, 98

3, 97

4, 96

5, 95

· ·

· ·

· ·

49, 61

50, 50

Se repitió 50 veces _

Estructura Repetitiva: (interrupciones)

break

Interrumpe la ejecución de un ciclo, y lo lleva al final.

Ejemplo 1:

```
# include (iostream.h)

int main() {
    for (int n=10; n>0; n--)
    {
        cout << n <<" ";
        if (n == 3) {
            cout <<"\nConteo interrumpido";
            break;}
    }
    return 0;}

```

10, 9, 8, 7, 6, 5, 4,
Conteo interrumpido

—

Estructura Repetitiva: (saltos)

continue

Produce un salto en todas las instrucciones y continua la ejecución hasta el final.

Ejemplo 1:

```
# include (iostream.h)
int main() {
    for (int n=10; n>0; n--)
    {
        if (n == 5) continue;
        cout << n <<" ";
    }
    cout << "\nFUEGO!!!\n";
return 0;}

```

10, 9, 8, 7, 6, 4, 3, 2, 1,
FUEGO!!!

—

Estructura Selectiva: (según caso)

switch

Nomenclatura:

```
switch (expresión) {  
    case constante 1:  
        lista de instrucciones 1  
        break;  
    case constante 2:  
        lista de instrucciones 2  
        break;  
    .  
    .  
    default:  
        lista de instrucciones por defecto  
}
```

- ◆ SWITCH toma el valor de la (expresión), y busca la constante equivalente y ejecuta la lista correspondiente. Si no lo encuentra ejecuta las instrucciones por defecto.

Estructura Selectiva: (según caso)

switch

Ejemplo:

```
switch (x)
{
    case 1:
        cout << "x vale 1";
        break;
    case 2:
        cout << "x vale 2";
        break;
    default:
        cout << "Valor de x desconocido";
}
```


FIN DE LA CLASE