

Unidad IV: Control de flujo

4.1 Estructuras secuenciales

La especificación formal de algoritmos tiene realmente utilidad cuando el algoritmo requiere una descripción más complicada que una lista de sencilla de instrucciones. Este es el caso cuando existe un número de posibles alternativas resultantes de la evaluación de una determinada condición.

Las estructuras selectivas se utilizan para tomar decisiones lógicas; de ahí que se suelen denominar también estructuras de decisión o alternativas.

En las estructuras selectivas se evalúan una condición y en función del resultado de la misma se realiza una opción u otra. Las condiciones se especifican usando expresiones lógicas. La representación de una estructura selectiva se hace con palabras en pseudocódigo (if, then, elseo bien en español si, entonces, si_no), con una figura geométrica en forma de rombo o bien con un triángulo en el interior de una caja rectangular, las estructuras selectivas o alternativas pueden ser:

- Simples
- Dobles
- Múltiples

4.2 Estructuras selectivas: simple, doble y múltiple

La estructura simple es si (if) con dos formatos: Formato Pascal, si-entonces (if-then) y en formato C, si(if) a estructura selectiva doble es igual que la estructura simple si al cual se le añade la cláusula si-no (else) a estructura selectiva múltiples según-sea (switch en lenguaje C, case en pascal)

4.3 Estructuras iterativas: repetir mientras, hasta, desde

CICLO FOR C++

Este ciclo es uno de los mas usados para repetir una secuencia de instrucciones, sobre todo cuando se conoce la cantidad exacta de veces que se quiere que se ejecute una instrucción simple o compuesta.

Su formato general es:

```
for (inicialización; condición; incremento)
```

```
{ instrucción(es); };
```

Funciona de la siguiente manera:

1. El valor inicial de la variable solo se ejecuta una vez al inicio del ciclo.
2. Se ejecuta la condición.
3. Si es verdadera pasará a ejecutar el cuerpo del for.
4. Se ejecuta el incremento o decremento de la variable.
5. Pasa a preguntar por la condición, resultando así un proceso recursivo.
6. Si la condición fuera falsa la ejecución pasará a la siguiente instrucción.

4.4 Diseño e implementación de funciones