

Bloques de decisión: un bloque de decisión es aquel que en base a una premisa es capaz de tomar una decisión y ejecutar un evento: **if, if-else, switch**.

Bloques de repetición: son aquellos que repiten una tarea cíclicamente hasta que se cumple una condición establecida por el usuario: **while, do-while, for, foreach**.

Ejemplos:

1. Ejemplo con if:

Hacer un programa que verifique que un número es par o impar:

```
Programa ParImpar
inicio
Variable numero;
LeerNumero numero;
Si numero != 0 entonces
    Variable residuo = numero%2;
    Si residuo == 0 entonces
        Imprimir "es par";
    Sino
        Imprimir "es impar";
    finSiSino
finSi
fin
```

2. Ejemplo con switch:

Hacer un programa que en base a un código de día L,M,Mi,J,V,S,D imprima el día de la semana:

Programa DiaDeSemana

Inicio

Variable código;

leerCadena código;

switch(codigo) entonces

case 'L':

imprimir "Lunes";

break;

case 'M':

imprimir "Martes";

break;

case 'Mi':

imprimir "Miércoles";

break;

case 'J':

imprimir "Jueves";

break;

case 'V':

imprimir "Viernes";

break;

case 'S':

imprimir "Sábado";

break;

case 'D':

```
        imprimir "Domingo";  
        break;  
    default:  
        imprimir "Ese código de día no existe";  
finSwitch  
fin
```

Bloques de repetición

El bloque while, traducido “mientras” a español es un bloque que se ejecuta hasta que se cumple una condición de verdad.

Ejemplo:

Hacer un programa que imprima 5 veces un texto cualquiera:

```
Programa ImprimirTextoRepetido  
Inicio  
Variable contador=0;  
Mientras contador < 5 hacer  
    imprimir "hola mundo";  
    contador = contador + 1;  
finMientras  
fin
```

El bloque do-while o “haga-mientras”, es un bloque de repetición único, este ejecuta sus instrucciones correspondientes y al final evalúa una condición. Este tipo de bloques son útiles para los problemas en los que se tienen que ejecutar acciones dependientes, por ejemplo, los menús de compra de paquetes de compañías telefónicas. Este tipo de bloques dependen de condiciones externas, como la acción de un usuario; por lo tanto, se debe de guiar al usuario en el proceso.

Ejemplo:

Hacer un programa que imprima un texto **hasta que el usuario ingrese -1**

Programa ImprimirHasta

Inicio

Variable numero;

Haga

Imprimir "Por favor ingrese un número: ";

leerNumero numero;

mientras numero != -1

finHagaMientras

fin

bloque for, o escrito "para" en español, este bloque de repetición se utiliza cuando se tiene un dominio conocido o finito de datos que no varían en el tiempo. Por ejemplo, el número de letras del abecedario.

Y la sintaxis es: **for**(variableContador ; condición ; paso¹){}

Ejemplo:

Hacer un programa que imprima del 1-10:

Programa imprimir110

Inicio

para(i=0;i<10;i++) **hacer**

imprimir i;

finPara

fin

¹ El paso es la cantidad de "saltos", puede ser de 1 en 1 o 2 en 2, etc. En general es de 1 en 1

bloque forEach, traducido como “para Cada”, es un bloque de repetición ampliamente utilizado para elementos con dimensiones variables en el tiempo. Por ejemplo: Listas de contactos, registros de bases de datos, datos agrupados insertados por el usuario y un sinfín de elemento con atributos y características afines.

Este bloque en muchos lenguajes, tienden a ser un atributo de los arreglos, en otros casos como Python es un for dinámicos.

Ejemplo:

Escriba un programa que imprima las siguientes listas: L1 = [1,2,3,4,5] y L2 = ['Juan','Pedro'];

Programa ImprimirListas

Inicio

Variable L1 = [1,2,3,4,5];

Variable L2 = ['Juan','Pedro'];

L1.forEach(subRutinaAnónima(valor){

 Imprimir valor;

});

L2.forEach(subRutinaAnónima(nombres){

 Imprimir nombres;

});

fin

Solución diagrama de pseudocódigo de el problema a ir a comprar un producto al supermercado

Programa ComprarProducto

Inicio

Variable dineroDisponible, categoriaProducto, precioProducto, nombreProducto, cantidad;

dineroDisponible = obtenerDineroDisponible();

categoriaProducto = obtenerCategoriaProducto();

cantidad = obtenerCantidadDeProducto();

variable categoría = caminarAlEstante();

si categoría == categoriaProducto **entonces**

nombreProducto = buscarProducto();

si nombreProducto.precio*cantidad > dineroDisponible **entonces**

buscarOtroProductoHastaEncontrarlo();

sino

comprarProducto();

finSiSino

sino

buscarEnOtroEstanteHastaEncontrarlo();

finSiSino

fin