INSTRUCCIONES BÁSICAS DE UN ALGORITMO

ING. GIANKARIS G. MORENO R., M.SC

ESTRUCTURAS REPETITIVAS

- Son aquellas que controlan la repetición de un conjunto de instrucciones mediante la evaluación de una condición.
- Están compuesta por:
  - ***Condición***: es la expresión lógica o relacional a ser evaluada y que determina la entrada o no al ciclo de una sentencia repetitiva.
  - ***Bloque de Instrucciones***: conjunto de instrucciones a ejecutarse.
  - ***Ciclo o Iteración***: el proceso de ejecución del bloque de instrucciones varias veces.

CONTADOR

- Es una variable cuyo valor se incrementa o decrementa en una cantidad constante cada vez que se produce un determinado suceso o acción.
- Los contadores se utilizan con la finalidad de contar sucesos o acciones internas de un bucle o ciclo.
- Es importante tener en cuenta que los contadores deben ser inicializados antes de utilizarse dentro del algoritmo.

FORMATO:

contador = contador + valor_constante;

- Ejemplo:

```
cont = cont + 1;
```

ACUMULADOR

- Es una variable que suma sobre sí misma un conjunto de valores, para de esta manera tener la sumatoria de todos ellos en una sola variable.
- También deben ser inicializados antes de utilizarse.
- La diferencia entre un contador y un acumulador es que el contador incrementa en un valor fijo, mientras que el acumulador lo hace en una cantidad variable.

FORMATO:

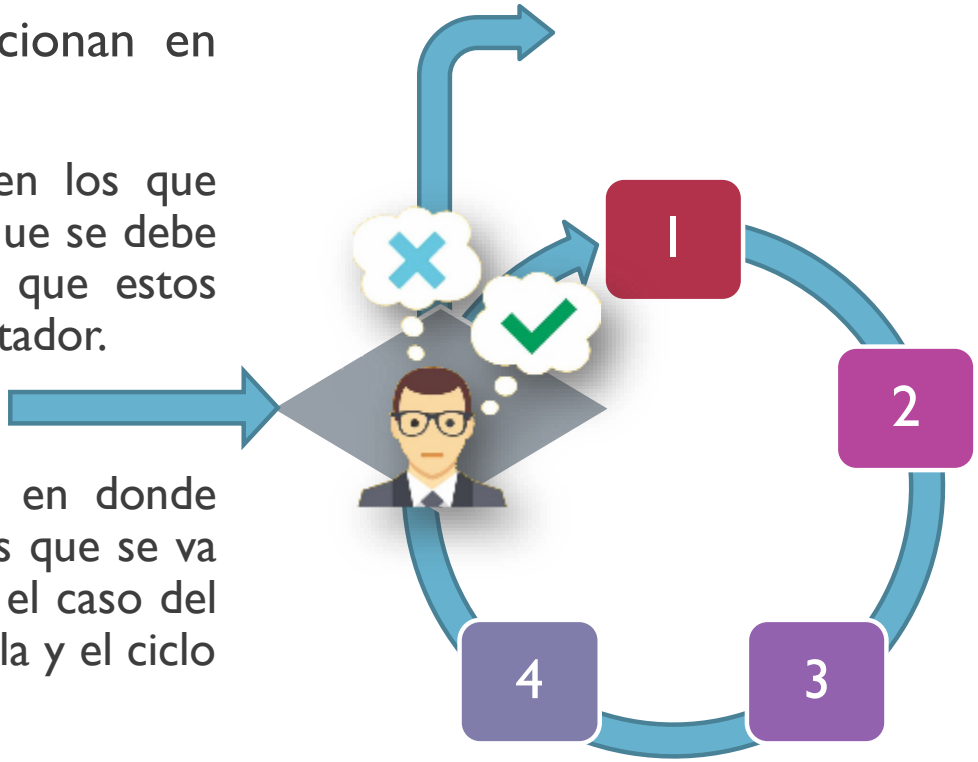
acumulador = acumulador + variable;

- Ejemplo:

```
acum = acum + nota;
```

ESTRUCTURAS REPETITIVAS

- Existen diversos tipos de estructuras repetitivas. Están las que funcionan en base a ciclos definidos y las que funcionan en base a ciclos indefinidos.
  - ***Ciclos Definidos***: son aquellos en los que conocemos la cantidad de veces que se debe ejecutar/repetir el ciclo. Se dice que estos ciclos son controlados por un contador.
  - ***Ciclos Indefinidos***: son aquellos en donde no se conoce la cantidad de veces que se va a ejecutar/repetir el ciclo. Este es el caso del ciclo controlado por valor centinela y el ciclo controlado por respuesta.



ESTRUCTURAS REPETITIVAS

- Existen diversas estructuras repetitivas:

Mientras

Hasta que

Para

MIENTRAS

- Esta estructura repetitiva evalúa una condición, mientras esa condición sea cierta, se ejecutará el bloque de instrucciones del ciclo. Finalizará su ejecución cuando la condición evaluada sea falsa.
- Formato:

```
mientras (condición)
{
    //bloque de instrucciones
}
```

MIENTRAS

- Ejemplo: Elabore un programa que genere e imprima los 10 primeros números enteros.

```
Algoritmo PrimerosNúmeros
{
    //Bloque Declarativas de Variables
    entero num;

    //Bloque de Instrucciones
    num = 0;  //se inicializa el contador
    imprimir ("Los 10 primeros números son: ");
    mientras (num < 10)
    {
        num= num + 1;
        imprimir (num);
    }
}
```

MIENTRAS

- Ejemplo: Elabore un programa que calcule la suma de los veinte primeros números pares e imprima el resultado.

```
Algoritmo NúmerosPares
{
        //Bloque Declarativas de Variables
        entero nump, cont, suma;

        //Bloque de Instrucciones
        nump = 0; // se inicializa el contador para generar los números pares
        cont = 0; // se inicializa el contador para controlar 20 números
        suma = 0; //se inicializa el acumulador
        mientras (cont < 20)
        {
                nump = nump + 2;
                suma = suma + nump;
                cont = cont + 1;
        }
         imprimir ("La suma de los 20 primeros números pares es: ", suma);
}
```

MIENTRAS

- Ejemplo: Elabore un programa que le permita al usuario sumar números enteros. El usuario determinará cuando finalizar y se mostrará el resultado de la sumatoria.

```
Algoritmo SumarNúmeros
{
        //Bloque Declarativas de Variables
        entero num, suma;
        caracter resp;

        //Bloque de Instrucciones
        resp = 's'; //se da valor inicial a la variable de control del ciclo
        suma = 0; //se inicializa el acumulador
        mientras (resp == 's' O resp == 'S')
        {
                imprimir ("Ingrese el número que desea sumar: ");
                leer (num);
                suma = suma + num;
                imprimir ("Desea continuar s/n: ");
                leer (resp);
        }
        imprimir ("La suma es = ", suma);
}
```

HASTA QUE (REPETIR)

- Esta estructura repetitiva evalúa una condición al final del ciclo, las instrucciones se ejecutarán hasta que la condición sea falsa.
- Esta instrucción permite que se itere por lo menos una vez, dado que su condición es evaluada al final.
- Formato:

```
repetir
{
      //bloque de instrucciones
} hasta que (condición);
```

HASTA QUE

- Ejemplo: Elabore un programa que genere e imprima los 10 primeros números enteros.

```
Algoritmo PrimerosNúmeros
{
    //Bloque Declarativas de Variables
    entero num;

    //Bloque de Instrucciones
    num = 0;  //se inicializa el contador
    imprimir ("Los 10 primeros números son: ");
    repetir
    {
        num= num + 1;
        imprimir (num);
    } hasta que (num < 10)
}
```

HASTA QUE

- Ejemplo: Elabore un programa que le permita al usuario sumar números enteros. El usuario determinará cuando finalizar y se mostrará el resultado de la sumatoria.

```
Algoritmo SumarNúmeros
{
    //Bloque Declarativas de Variables
    entero num, suma;
    caracter resp;

    //Bloque de Instrucciones
    suma = 0; //se inicializa el acumulador
    repetir
    {
        imprimir ("Ingrese el número que desea sumar: ");
        leer (num);
        suma = suma + num;
        imprimir ("Desea continuar s/n: ");
        leer (resp);
    } hasta que (resp == 's' O resp == 'S');
    imprimir ("La suma es = ", suma);
}
```

HASTA QUE

- Ejemplo: Elabore un programa que simule un sensor de velocidad mediante el ingreso de velocidades, el mismo debe indicarle al usuario "Exceso de velocidad" cuando la misma alcance o exceda los 110km/h.

```
Algoritmo SensorVelocidad
{
        //Bloque Declarativas de Variables
        entero vel;

        //Bloque de Instrucciones
        repetir
        {
                imprimir ("Sensando la velocidad:  ");
                leer (vel);
        } hasta que (vel < 110);
        imprimir ("Exceso de velocidad");
}
```