Estructura de control *Mientras*



Definición y sintaxis

Conocimientos previos:
- Manejo de conceptos básicos de los algoritmos computacionales
- Manejo de programación estructurada

Duración: 5 horas
Comunidad: estudiantes a nivel licenciatura o técnico de formación básica en Ingeniería y Ciencias Computacionales
Grado de dificultad: fácil

Universidad Veracruzana

Propósito del tema

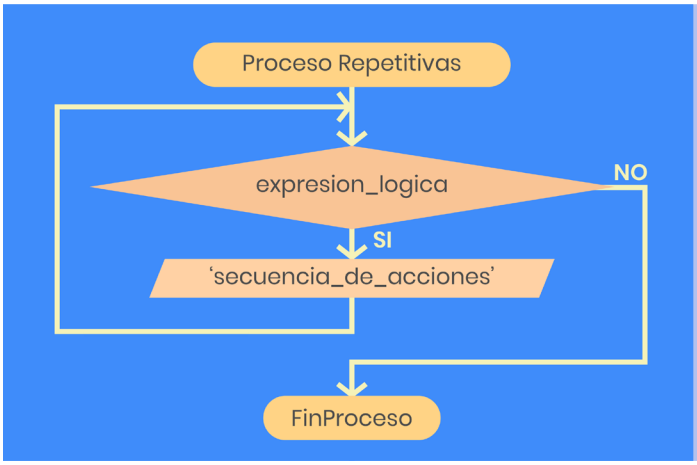
Analizar la estructura de control *Mientras* para su correcta implementación.

Sentencias repetitivas

Las estructuras de control repetitivas son aquellas que permiten ejecutar un conjunto de instrucciones varias veces, de acuerdo al valor que genere la expresión relacional y/o lógica. Esta estructura repetitiva se utiliza principalmente cuando no se conoce el número de veces que las acciones o el ciclo deben repetirse, aunque también puede utilizarse en otros casos.

Las estructuras repetitivas, permiten repetir una acción (o grupo de acciones) varias veces. Existen tres tipos de estructuras repetitivas: mientras, desde-hasta y repetir-hasta.

En los diagramas de flujo, un ciclo se representa de la siguiente manera:



Aquí puede observar que las líneas de flujo, indican el orden a seguir y según el valor de la condición, continuará ejecutándose el mismo conjunto de instrucciones o saldrá del ciclo.

Las estructuras de control repetitivas utilizan dos tipos de variables.

Contadores

Un contador es una variable de tipo entero que incrementa o decrementa su valor de forma ¬¬para contar el número de veces que se ejecuta una acción o para contar el número de veces que se cumple una condición (expresión relacional/lógica).

Por ejemplo, si se desea sumar los números del 1 al 5, se necesitará una variable que genere esos números, es decir que empiece en 1 y llegue hasta el 5.

La variable que cumple el rol de contador aparece tanto a la izquierda como a la derecha, por la propiedad destructiva de la asignación; así tomará el valor anterior, le adicionará o reducirá el valor constante y asignará el nuevo valor.

Acumuladores

Un acumulador es una variable numérica, que incrementa o decrementa su valor de forma NO CONSTANTE y requiere ser inicializada. Como su nombre lo indica se utilizan para acumular valores en una sola variable, ya sea de suma o producto. Por lo tanto, existen dos modos de inicialización:

- Para Suma: Inicializar en 0
- Para Producto: Inicializar en 1

Esto con el objetivo de no alterar los valores de las respectivas operaciones.

Ejemplo: Si se desea conocer el acumulado de los pagos realizados a un grupo de empleados, se necesitará una variable que vaya sumando los sueldos de cada empleado, se requiere una variable que permita calcular el acumulado.

Para identificar qué procesos van dentro de una instrucción repetitiva debe preguntarse lo siguiente:

- De las acciones requeridas para la solución de este problema, ¿cuáles se realizan VARIAS VECES?
- Todas las respuestas a esta pregunta corresponden a las instrucciones que van dentro de la sentencia repetitiva.

Ciclo Mientras (While)

El ciclo *while* permite ejecutar n veces un conjunto de instrucciones siempre y cuando la condición se cumpla o sea verdadera.

Esta estructura de control repite un conjunto de instrucciones mientras una condición se cumpla, en cuanto la condición no se cumple el ciclo deja de ejecutarse. En el caso de que la condición se evalúe por primera vez como falsa, el ciclo no será ejecutado. Por ejemplo, pensemos en un caso de la vida real, donde un coche avanza por una avenida, mientras tenga gasolina, el coche dejará de avanzar cuando ésta se agote.



Esta estructura repetitiva se utiliza principalmente cuando no se conoce el número de veces que las acciones o el ciclo deben repetirse, aunque también puede utilizarse en otros casos.

Si la condición siempre es verdadera o siempre se cumple este ciclo, puede crear un bucle que nunca termine, por eso hay que tener en cuenta que en algún momento del ciclo la condición no debe cumplirse (debe volverse falsa) para que el ciclo termine, caso contrario el ciclo se vuelve infinito

En la estructura de control se distinguen dos partes:
Ciclo: Conjunto de instrucciones que se ejecutarán repetidamente.
Condición de terminación del ciclo: La evaluación de esta condición determinará la finalización del ciclo.

Elementos indispensables en la estructura de control *Mientras*

1. **Instrucción inicial** — a) Debe existir una **instrucción inicial** (antes de la estructura de control) que haga válida la condición a evaluar para que al menos se ejecute una vez el ciclo.
2. **Condición** — b) Debe existir una **condición** que se evaluará desde el inicio y cuando no sea válida el ciclo dejará de ejecutarse.
3. **Instrucción** — c) Debe existir una **instrucción** dentro del ciclo que afecte la condición **para evitar que el ciclo se ejecute de manera indefinida.**

```
condición=verdadera (a)        ← Condición de terminación del ciclo
Mientras <condición (b)> hacer
        <instrución 1...
         instrucción 2...
         Instrucción que afecte la condición (c)>   ← Ciclo
Fin mientras.
```

```
Mientras <condición> hacer
  <instrucción 1>
  <instrucción 2>
Fin mientras
```

Las letras marcadas en rojo se identifican como palabras reservadas y son necesarias para considerar correcta la sintaxis de esta estructura, la falta de cualquier de ellas no permitirá la ejecución del algoritmo. En PSeInt se puede omitir la palabra reservada hacer y no afectará la ejecución del algoritmo.

Imagina que debes calcular el total de gastos de un estudiante en un mes y no sabes con exactitud cuántos gastos fueron.

```
Inicio de algoritmo
   sumadegastos = 0;
   gasto = 0;
    respuesta = "si"; //(Instrucción que hace válida la condición a evaluar, la condición a evaluar es: Mientras respuesta = "si")
   Mientras respuesta == "si" Hacer
      Escribir "Introduce gasto";
      Leer gasto;
      sumadegastos = sumaDeGastos + gasto;
      Escribir "Tiene otro gasto?";
        Leer respuesta; //(Instrucción que afecta la condición, en cuanto la respuesta sea diferente a "si", el ciclo dejará de ejecutarse)
   Fin mientras
   Escribir "El total de gastos es:", sumaDeGastos;
Fin de algoritmo
```

Imagina que deseas saber el total de ingreso por ventas en un día en una tienda de abarrotes y no sabes con exactitud cuántas ventas se hicieron.

```
Inicio del algoritmo
   totalIngresos=0;
    respuesta= "si"; //(Instrucción que hace vá lida la condición a evaluar, la condición a evaluar es: Mi entras respuesta= "si")
   Mientras respuesta = "si" hacer
      Escribir "Introduce la venta";
      Leer venta;
      totalIngresos = totalIngresos + venta;
      Escribir "¿Tiene otra venta por registrar?";
      Leer respuesta;
```

```
    Fin mientras
    Escribir “El ingreso total por ventas del día fue de:”, totalIngresos
Fin algoritmo
```

Referencias bibliográficas

Cairo, O. (2005) Metodología de la programación: algoritmos, diagramas de flujo y programas. Alfaomega. México.

Joyanes, L. (2008) Fundamentos de la programación. McGraw-Hill. España