

Taller de Programación

Clase 2 - Estructuras de control

Mgs. Lic. Marcos Prunello

2017-08-27

Novedades: Práctica N° 2 y Actividad Evaluativa N° 1

- Resolver y entregar los ejercicios 9, 10 y 11 de la práctica N° 2.
- Fecha de entrega: 18/9/17.
- Grupos de 3 integrantes.
- Más instrucciones en el Campus Virtual.

Declaración de variables y constantes

- Las variables y constantes que emplearemos en nuestro algoritmo deben ser declaradas al inicio del mismo.
- Declarar significa nombrarlas y especificar el tipo de datos que contendrán (numérico, carácter, lógico)
- Muchos lenguajes de programación utilizan esta declaración para reservar en la memoria de la computadora un espacio para almacenar la información correspondiente de manera adecuada.

Algunos puntos que quedaron pendientes

Entrada y salida de información

- Acción LEER: se usa para requerir que un usuario provea información y registrar la misma como un valor que debe ser asignado a una variable.
- Acción ESCRIBIR: se usa para mostrar un resultado en un mensaje.
 - Las palabras o frases literales que se desean mostrar en el mensaje deben estar encerradas entre comillas porque son cadenas de texto.
 - Si se desea mostrar el valor de una variable se debe escribir su identificador sin comillas.

Algunos puntos que quedaron pendientes

Ejemplo

ALGORITMO: "Calcular área de un círculo"

COMENZAR

CONSTANTE numérica pi

VARIABLE numérica radio, area

pi \leftarrow 3.1416

LEER radio

area \leftarrow pi * radio ** 2

ESCRIBIR "El área del círculo es ", area

FIN

- Las estructuras de control son comandos especiales que alteran o controlan el flujo de la sucesión de acciones que componen el algoritmo.
- Hay estructuras secuenciales, condicionales e iterativas.
- Las **estructuras secuenciales** están compuestas por un número definido de acciones que se ubican en un orden específico y se suceden una tras otra, como los ejemplos vistos.

- El curso de acción depende del resultado de la comparación una variable con otro valor, que puede ser una constante u otra variable.
- Existen tres tipos de estructuras condicionales: **simples**, **dobles** y **múltiples**.

Estructuras condicionales simples

Representan una toma de decisión y se describen con la siguiente sintaxis:

```
SI <condición> ENTONCES  
    Acción/es  
FIN SI
```


Estructuras condicionales simples

Ejemplo:

ALGORITMO: "Determinar mayoría de edad"

COMENZAR

VARIABLE numérica edad

LEER edad

SI edad \geq 18 ENTONCES

 ESCRIBIR "Es mayor de edad"

FIN SI

FIN

Estructuras condicionales dobles

```
SI <condición>  
    ENTONCES  
        Acción/es  
SI NO  
    Acción/es  
FIN SI
```

Estructuras condicionales dobles

Ejemplo:

```
ALGORITMO: "Determinar mayoría de edad"
COMENZAR
    VARIABLE numérica edad
    LEER edad
    SI edad >= 18
        ENTONCES
            ESCRIBIR "Es mayor de edad"
        SI NO
            ESCRIBIR "Es menor de edad"
    FIN SI
FIN
```

Estructuras condicionales múltiples o anidadas

```
SI <condición>
  ENTONCES
    Acción/es
  SI NO
    SI <condición>
      ENTONCES
        Acción/es
      SI NO
        Acción/es
    FIN SI
  FIN SI
```

- Ejemplo: en relación a lo anterior, una exhibición permite el ingreso a mayores edad pero los menores sólo pueden hacerlo si tienen la autorización de un adulto a cargo:

Estructuras condicionales múltiples o anidadas

ALGORITMO: "Determinar ingreso"

COMENZAR

VARIABLE numérica edad

VARIABLE lógica autorizado

LEER edad

SI edad \geq 18

ENTONCES

 ESCRIBIR "Permitir ingreso"

SI NO

 LEER autorizado

 SI autorizado = VERDADERO

 ENTONCES

 ESCRIBIR "Permitir ingreso"

 SI NO

 ESCRIBIR "Denegar ingreso"

 FIN SI

FIN SI

FIN

Estructuras de control iterativas: número fijo de iteraciones

- Se aplican cuando se conoce de antemano el número exacto de veces que se debe repetir una secuencia de acciones.
- Por ejemplo, si deseamos mostrar en pantalla la tabla de multiplicar del número 8 completa podríamos hacer:

ALGORITMO: "Mostrar tabla del 8"

COMENZAR

```
VARIABLE numérica resultado
resultado <- 8 * 1
ESCRIBIR "8 x 1 = ", resultado
resultado <- 8 * 2
ESCRIBIR "8 x 2 = ", resultado
resultado <- 8 * 3
ESCRIBIR "8 x 3 = ", resultado
...
resultado <- 8 * 10
ESCRIBIR "8 x 10 = ", resultado
```

FIN

Estructuras de control iterativas: número fijo de iteraciones

- Esto puede resumirse así:

ALGORITMO: "Mostrar tabla del 8"

COMENZAR

VARIABLE numérica resultado

PARA i DESDE 1 HASTA 10 HACER

 resultado \leftarrow 8 * i

 ESCRIBIR "8 x ", i, " = ", resultado

FIN PARA

FIN

Estructuras de control iterativas: número fijo de iteraciones

Sintaxis:

```
PARA <variable> DESDE <inicio> HASTA <fin> CON PASO <paso> HACER  
    Acción/es  
FIN PARA
```


Estructuras de control iterativas: número indeterminado de iteraciones

- Para cuando se necesita repetir un bloque de acciones sin conocer con exactitud cuántas veces.
- Las iteraciones pueden continuar **mientras que** o **hasta que** se verifique alguna condición.

Mientras que

- El conjunto de sentencias se repite mientras que se siga evaluando como VERDADERO una condición declarada al inicio del bloque.
- Cuando la condición ya no se cumple, el proceso deja de ejecutarse.
- La sintaxis es:

```
MIENTRAS QUE <condición> HACER  
    Acción/es  
FIN MIENTRAS
```

Mientras que

Ejemplo:

ALGORITMO: "Múltiplos de 8 menores a 150"

COMENZAR

VARIABLE numérica multiplo

multiplo <- 8

MIENTRAS QUE multiplo < 150 HACER

 ESCRIBIR multiplo

 multiplo <- multiplo + 8

FIN MIENTRAS

FIN

Observaciones:

- La evaluación de la condición se lleva a cabo antes de cada iteración, incluyendo la primera. Si la condición es FALSO inicialmente, entonces las acciones en el cuerpo de la estructura no se ejecutan nunca.
- La evaluación de la condición **sólo** se lleva a cabo al inicio de cada iteración. Si la condición se vuelve FALSO en algún punto durante la ejecución de un bloque, el programa no lo nota hasta que se termine de ejecutar el bloque y la condición sea evaluada antes de comenzar la próxima iteración.

Hasta que

- Repite el bloque de acciones hasta que se cumpla una condición, es decir, se ejecuta mientras que dicha condición sea evaluada como FALSA.
- La sintaxis es:

REPETIR

Acción/es

HASTA QUE <condición>

Hasta que

Ejemplo:

ALGORITMO: "Múltiplos de 8 menores a 150"

COMENZAR

VARIABLE numérica multiplo

multiplo <- 8

REPETIR

 ESCRIBIR multiplo

 multiplo <- multiplo + 8

HASTA QUE multiplo >= 150

FIN

Hasta que

Observación:

- El bloque de acciones se ejecuta al menos una vez, dado que la condición se evalúa al final.

```
"_._"
```