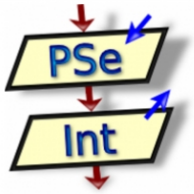
CURSO DE PROGRAMACIÓN FULL STACK

ESTRUCTURAS DE CONTROL CON PSEINT

ESTRUCTURAS REPETITIVAS



EGG

GUÍA DE ESTRUCTURAS DE CONTROL

En la guía anterior vimos las estructuras de control secuenciales y selectivas. En esta guía vamos a continuar con las estructuras de control pero ahora sumando las estructuras repetitivas.

ESTRUCTURAS REPETITIVAS

Durante el proceso de creación de programas, es muy común, encontrarse con que una operación o conjunto de operaciones deben repetirse muchas veces. Para ello es importante conocer las estructuras de algoritmos que permiten repetir una o varias acciones, un número determinado de veces.

Las estructuras que repiten una secuencia de instrucciones un número determinado de veces se denominan *bucles*, y se denomina *iteración* al hecho de repetir la ejecución de una secuencia de acciones.

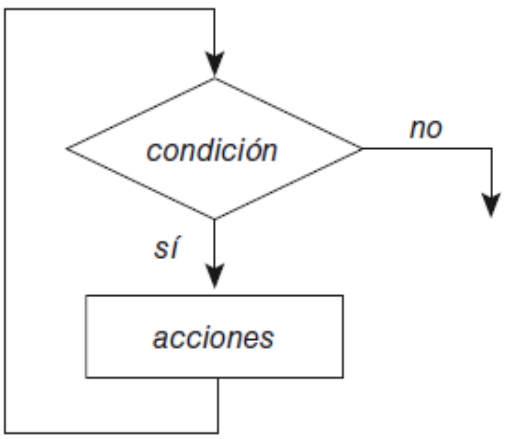
Todo bucle tiene que llevar asociada una condición, que es la que va a determinar cuándo se repite el bucle y cuando deja de repetirse.

Hay distintos tipos de bucles:

- *Mientras*
- *Hacer Mientras*
- *Para*

ESTRUCTURA MIENTRAS

Esta estructura repetitiva *Mientras*, es en la que el cuerpo del bucle se repite siempre que se cumpla una determinada *condición*. Cuando se ejecuta la instrucción mientras, la primera cosa que sucede es que se evalúa la condición (una expresión lógica). Si se evalúa *falsa*, no se toma ninguna acción y el programa prosigue con la siguiente instrucción. Si la expresión lógica es *verdadera*, entonces se ejecuta el cuerpo del bucle, después de lo cual se evalúa de nuevo la expresión lógica. Este proceso se repite una y otra vez mientras la expresión lógica (condición) sea verdadera, para salir del bucle la condición debe ser falsa.



Pseudocódigo en PSeInt:

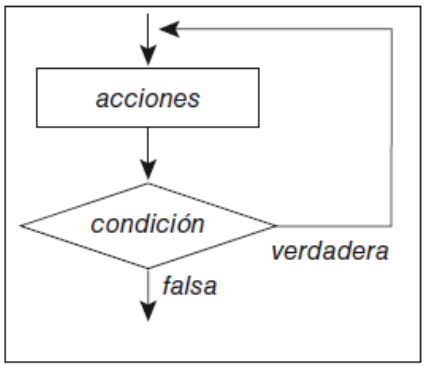
```
Mientras expresion_logica Hacer
      secuencia_de_acciones
Fin Mientras
```

Regla práctica

Las pruebas o test en las expresiones lógica es conveniente que sean mayor o menor que en lugar de pruebas de igualdad o desigualdad. En el caso de la codificación en un lenguaje de programación, esta regla debe seguirse rígidamente en el caso de comparación de números reales, ya que como esos valores se almacenan en cantidades aproximadas las comparaciones de igualdad de valores reales normalmente plantean problemas. Siempre que realice comparaciones de números reales use las relaciones <, <=, > o >=.

ESTRUCTURA HACER- MIENTRAS

Esta estructura es muy similar a la anterior, sólo que a diferencia del *Mientras* el contenido del bucle *Hacer-Mientras* se ejecuta siempre al menos una vez, ya que la evaluación de la condición lógica se encuentra al final del bucle. De esta forma garantizamos que las acciones dentro de este bucle sean llevadas a cabo al menos una vez, incluso aunque la expresión lógica sea falsa.



Pseudocódigo en PSeInt:

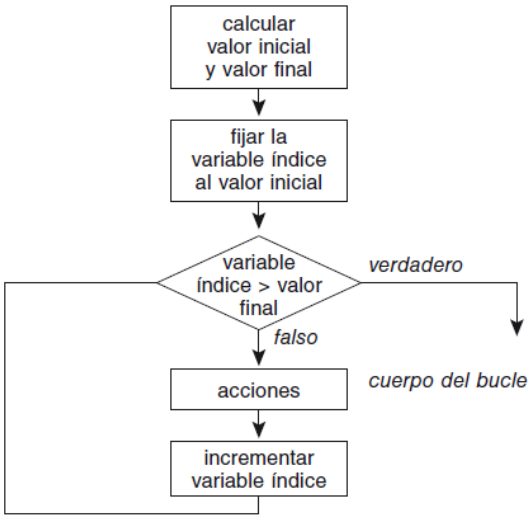
```
Hacer
      secuencia_de_acciones
Mientras Que expresion_logica
```

Regla práctica

El bucle *hacer-mientras* se termina de ejecutar cuando el valor de la condición es falso. La elección entre un bucle mientras y un bucle hacer-mientras depende del problema de cómputo a resolver. En la mayoría de los casos, el bucle mientras es la elección correcta. Por ejemplo, si el bucle se utiliza para recorrer una lista de números (o una lista de cualquier tipo de objetos), la lista puede estar vacía, en cuyo caso las sentencias del bucle nunca se ejecutarán. Si se aplica un bucle hacer-mientras nos conduce a un código de errores.

ESTRUCTURA PARA

La estructura *Para* es un poco más compleja que las anteriores y nos permite ejecutar un conjunto de acciones, para cada paso de un conjunto de elementos. Su implementación depende del lenguaje de programación, pero en términos generales podemos identificar tres componentes: la *inicialización*, *finalización* y el *incremento*.
La estructura *Para* comienza con un valor inicial de una variable llamada índice y las acciones especificadas se ejecutan x cantidad de veces, hasta que el valor índice llegue al valor final, *a menos que el valor inicial sea mayor que el valor final*. La variable índice se incrementa en uno y si este nuevo valor no excede al final, se ejecutan de nuevo las acciones. Por consiguiente, las acciones específicas en el bucle se ejecutan para cada valor de la variable índice desde el valor inicial hasta el valor final con el incremento de uno en uno.



Pseudocódigo en PSeInt:

```
Para variable_numerica<-valor_inicial Hasta valor_final Con Paso paso Hacer
      secuencia_de_acciones
Fin Para
```

El incremento de la variable índice (variable_numerica) siempre es 1 si no se indica expresamente lo contrario en el valor de *con paso*. Dependiendo del tipo de lenguaje, es posible que el incremento sea distinto de uno, positivo o negativo. La variable índice o de control (variable_numerica) normalmente será de tipo entero y es normal emplear como nombres las letras i, j, k.

Si el valor_inicial de la variable índice es menor que el valor_final, los incrementos, es decir los pasos, deben ser positivos, ya que en caso contrario la secuencia de acciones no se ejecutaría. De igual modo, si el valor_iniciales mayor que el valor_final, el paso debe ser en este caso negativo, es decir, decremento.