

Diseño estructurado de algoritmos

Sesión 9: Estructuras Algorítmicas

Contextualización

¿Qué componentes crees que permiten el desarrollo de un algoritmo más complejo?

Las estructuras algorítmicas son componentes básicos para el desarrollo de un programa, si se sabe combinar estas estructuras se puede llegar a un diseño más elaborado y complejo.

Las estructuras son un conjunto de formas de trabajo que permiten el manejo de variables para la realización de procesos con los que se pueda llegar a la solución de problemas. Aplicarlas de manera adecuada permitirán expresar las ideas, pero con una cierta restricción para la lógica de los algoritmos.

El hecho de que se crearan las estructuras fue para estandarizar, lo que se pretendía era que un programa pudiera ser entendido fácilmente por cualquier tipo de persona, porque si cada persona usara su lógica existirían programas confusos que solamente entendería su creador.

Esa es la razón por la cual se uniformó la lógica para el desarrollo de algoritmos.



Introducción

¿Te gustaría conocer la metodología necesaria para realizar una estructura basada en el desarrollo de alternativas?

Para estructurar un programa se debe seguir una metodología que permita conseguir un propósito, la metodología se caracteriza por una serie de tareas llevadas a cabo.

Esta serie de tareas guían al programador a la realización de un código, por lo que resulta importante conocer las estructuras y todos los componentes necesarios para su desarrollo.

Por este motivo, esta sesión pretende dar a conocer la estructura algorítmica basada en el desarrollo de alternativas y los conceptos necesarios para su realización.

Por ejemplo, uno de los términos muy utilizados en las estructuras algorítmicas es el de asignación, por lo cual es importante poder dominarlos todos para su mayor comprensión.



Estructuras Algorítmicas

¿Sabes cómo se les denomina a los datos que el usuario introduce en un programa?

Los datos que se obtienen de un usuario para ser procesados en la ejecución de un programa son llamados “datos de Entrada”. Estos datos deben ser asignados a variables. La operación que realiza el programa es: lee un dato y lo asigna a una variable.



El procedimiento que ejecuta un programa denominado lectura, consiste recibir un dato o valor que el usuario teclee. Para que la computadora realice el proceso de lectura de un dato ingresado se utiliza la instrucción *leer* o *ingresar* y se representa en un pseudocódigo de la siguiente forma:

Leer a

Donde “a” representa la variable donde el programa guardo el dato ingresado por el usuario.

Datos de entrada

Por ejemplo

- Leer Edad



Esta instrucción le pide al usuario que ingrese su edad, este dato se asignará en la variable “Edad”. Como recordarás en la sesión pasada vimos que el término asignar se refiere a el valor que se le da a un identificador o variable.

Ahora que sabes cómo se les llama a los datos que el usuario introduce,

¿Cuáles crees que son los datos de Salida?

Datos de Salida

Cuando el programa realizó un proceso, los datos que le muestra al usuario son llamados datos de salida. Puede ser el resultado de una operación, un mensaje de error, etc.

Para que la computadora arroje un resultado se utiliza la instrucción *imprimir* y se acompaña de comillas si lo que se desea mostrar es una cadena de caracteres.

Imprimir "El valor de la suma es:", Suma

Esta instrucción muestra el mensaje que esta entre comillas y luego el valor que se guardó en la variable Suma. La coma permite separar el mensaje y el resultado, por lo que el usuario podrá ver el resultado impreso en la pantalla de la siguiente forma:

El valor de la suma es: 5



Datos de Salida

Una vez que se ha comprendido lo que se desea que realice el programa, es necesario definir:

- Datos de entrada.
- Datos de salida

Para que arroje los datos de salida, la computadora realiza un proceso que se podría representar de la siguiente forma:



Estructuras Condicionales

- ***¿Conoces las estructuras de selección?***



Las estructuras de selección se construyen basándose en instrucciones condiciones, generando alternativas. Estas estructuras controlan el flujo del programa.

Las estructuras condicionales permiten realizar preguntas sobre el resultado que se desea que arroje el programa, actuando según la respuesta obtenida. Las respuestas solo muestran dos caminos diferentes: Si o No. Si la instrucción es verdadera, se ejecutan unas líneas de código declaradas al principio del programa, si es falsa, esas líneas no se ejecutan.

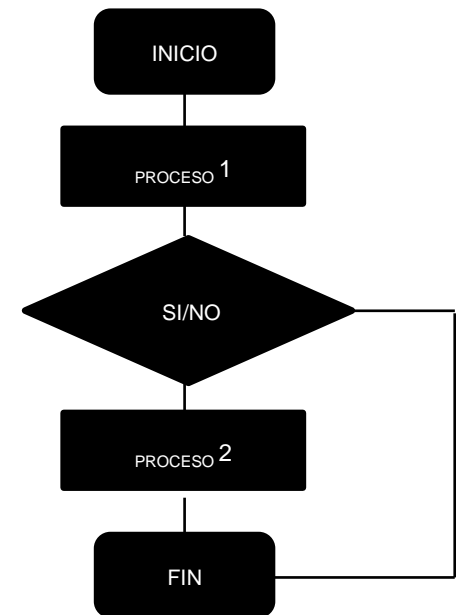
Estructuras Condicionales Simples

Existen dos tipos de estructuras condicionales:

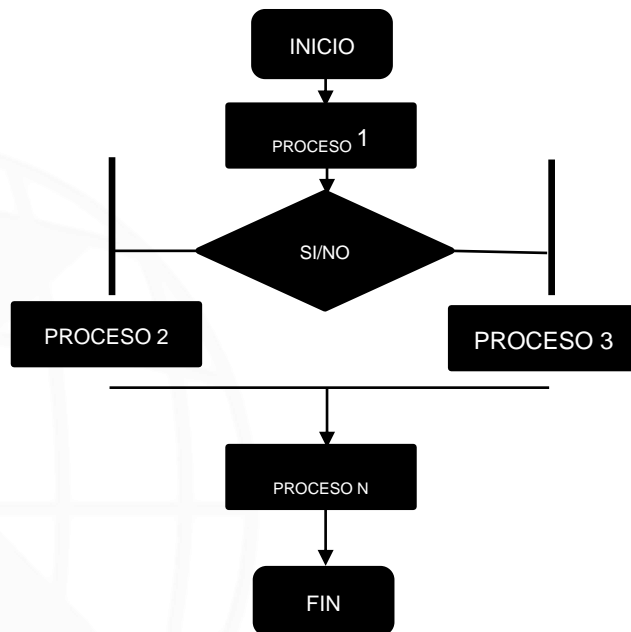
1.- Simples:

Las estructuras simples permiten tomar una decisión.

Por ejemplo la imagen de la derecha nos indica Si se cumple la condición entonces ejecuta otro proceso, sino brinca al final.



Estructuras Condicionales Dobles



2.- Dobles:

Las estructuras dobles permiten elegir entre dos opciones según se cumpla la condición. A diferencia de las estructuras simples, esta estructura en cualquiera de los resultados se a Si o No, se realiza un proceso.

Conclusión

En nuestra vida siempre estamos llenos de decisiones, pero la forma de solucionarlo no puede ser al azar, debe haber un análisis que permita escoger el mejor camino para llegar a la solución.

Los programadores al momento de generar el diseño de su algoritmo deben de saber manejar estructuras que hagan más fácil la toma de decisiones.

Es habitual que, cuando se está en búsqueda de una solución existan diferentes acciones que deban ser procesadas, para saber cuál debe ejecutarse existe una condición que determina el camino que debe seguirse.



Conclusión

Las estructuras condicionales permiten realizar preguntas sobre el resultado que se desea que arroje el programa para después actuar según la respuesta obtenida. Recordando que solo hay dos condiciones CIERTO o FALSO.

- ¿Crees que esta estructura te funcione para solucionar problemas cotidianos?
- ¿Conoces algún otro tipo de estructura condicional que permita la comparación de variables contra distintos posibles resultados?

Para responder a estas preguntas en la siguiente sesión se detallará más el tema de estructuras condicionales y sus tipos.



Para aprender más

¿Quieres ver un ejemplo aplicado al tema de esta sesión?

El siguiente video te permitirá entender con base a ejemplos cómo se aplican las condiciones.

- Gonzales Ivan.2013. *Estructuras Condicionales en Algoritmos*. Consultado el 21 de Marzo del 2013, de: <http://www.youtube.com/watch?v=-PKUrPBUE20>

Por medio de este enlace reforzarás lo aprendido en la sesión y analizarás ejemplos que contengan su pseudocódigo y el diagrama de flujo.

- Salleg Boris.2012. *Condicionales*. Consultado el 21 de Marzo del 2013, de: <http://www.slideshare.net/BorisSalleg/condicionales-12142225>

Referencias

- Trejos Buriticá Omar Iván. (s/a). *La Esencia de la Lógica de Programación. Capítulo 4, Estructuras Básicas y Técnicas para Representar Algoritmos*. Consultado el 21 de Marzo de 2013, de <http://archive.p2pu.org/sites/archive.p2pu.org/files/document/27233-clase-4-semana-4/cap04.pdf>
- s/a.2012.*Estructuras Algorítmicas Condicionales*. Consultado el 21 de Marzo del 2013, de: <http://estructurascondicionales.blogspot.mx/>
- Vázquez Gómez Juan Bernardo. 2012. Red Tercer Milenio. *Análisis y diseño d algoritmos*. pp 76-78. Consultado el 21 de Marzo del 2013, de: http://www.aliatuniversidades.com.mx/bibliotecasdigitales/pdf/sistemas/Analisis_y_disenio_de_algoritmos.pdf



UNID[®]

UNIVERSIDAD
INTERAMERICANA
PARA EL DESARROLLO