

Módulo formativo: Programación y base de datos

Objetivo: Desarrollar sistemas informáticos con lenguajes de programación y base de datos, aplicando diferentes metodologías según los requerimientos de funcionalidad.

Ficha N.º BT5011	
Tema: Datos y expresiones	Contenido procedimental: Seleccionar las técnicas adecuadas para solucionar problemas mediante la codificación y representación de algoritmos y programas.
Contenidos	Actividades de aprendizaje
<p>La palabra ‘algoritmo’ es una de las más usadas en la actualidad, rodeados como estamos de sistemas de inteligencia artificial. Pero ¿qué es un algoritmo en realidad?</p> <p>La computadora, por muy mágica que parezca, no es más que una máquina matemática. Todos los prodigios que realiza los hace por medio de cálculos, siguiendo las instrucciones de los programas. Ese conjunto de instrucciones detalladas es lo que conocemos como algoritmo.</p> <p>Declaración de variables Durante la ejecución de un algoritmo, la computadora crea y modifica lo que llamamos «variables». Una variable es un espacio en memoria que tiene un nombre y almacena valores numéricos. Por ejemplo, si el programador escribe algo como: $a = 3$ le está diciendo a la computadora que reserve un espacio en memoria, que le ponga el nombre “a” y almacene el número 3 en él.</p> <p>Sobrescribir variables Una sentencia similar a la siguiente: $a = a + 8$ tiene el efecto de sumar un 8 a lo que está guardado en la variable “a”. El nuevo valor guardado será el 11.</p> <p>Bucle for A veces queremos que la computadora ejecute muchas veces una misma acción. Por ejemplo, que realice cinco veces la operación $a = a + 8$ Para lograrlo, podemos explicarle a la computadora: $a = a + 8$ $a = a + 8$ $a = a + 8$ $a = a + 8$</p>	<p>Tareas: Considerar el siguiente flujograma, que calcula el costo del pasaje de acuerdo con la edad del pasajero (simplificado, no considera la condición de discapacidad o estudiante):</p> <pre> graph TD INICIO[INICIO] --> INGRESAR[INGRESAR EDAD (e) del PASAJERO] INGRESAR --> D1{e < 12} D1 -- Sí --> P1[p = 0,17] D1 -- No --> D2{e > 65} D2 -- Sí --> P2[p = 0,35] D2 -- No --> P2 P1 --> TERMINAR[TERMINAR] P2 --> TERMINAR </pre> <p>Indicar con un lápiz el camino que seguirá el algoritmo en los siguientes casos:</p> <ol style="list-style-type: none"> Niño de 10 años Adulto de 35 Adulto mayor de 80 <p>Escribir (en pseudocódigo) un algoritmo para calcular el factorial de un número. El factorial de 5, por ejemplo, es: $5 * 4 * 3 * 2 * 1 = 120$ El factorial de 3 es: $3 * 2 * 1 = 6$ Este algoritmo podría empezar con (para el primer caso): $n = 5$...</p>

$a = a + 8$

Obviamente, debe haber una manera más práctica de hacerlo. Por eso se crearon las «sentencias de control», como el bucle for. Una manera de escribir una sentencia de control con un bucle for sería:

for i = 1:5

$a = a + 8$

Cuando recibe esta instrucción, la computadora crea internamente la variable «i», a la cual asigna los valores 1, 2, 3, 4 y 5 en cada paso o «iteración» (es como si contara el número de veces que va realizando la operación).

Pruebas de escritorio

Cuanto más complejo es un algoritmo, más difícil es predecir cómo se comportará. Por ejemplo, ¿qué quedará almacenado en la variable «b» cuando la computadora ejecute el siguiente algoritmo?

b = 5

for i = 1:3

$b = b + i$

La prueba de escritorio nos servirá para entender los pasos que dará la computadora y puede ponerse en forma de una tabla:

b	i
5	1
6	2
8	3
11	

Bucle while

Recordando que la palabra inglesa ‘while’ significa ‘mientras’, es fácil intuir que la computadora ejecutará lo que esté dentro del while *mientras* se cumpla alguna condición. Por ejemplo, al seguir la instrucción:

c = 5

while c > 1

$c = c/2$

la computadora ejecutará el $c = c/2$ mientras c sea mayor que 1.

La prueba de escritorio quedaría:

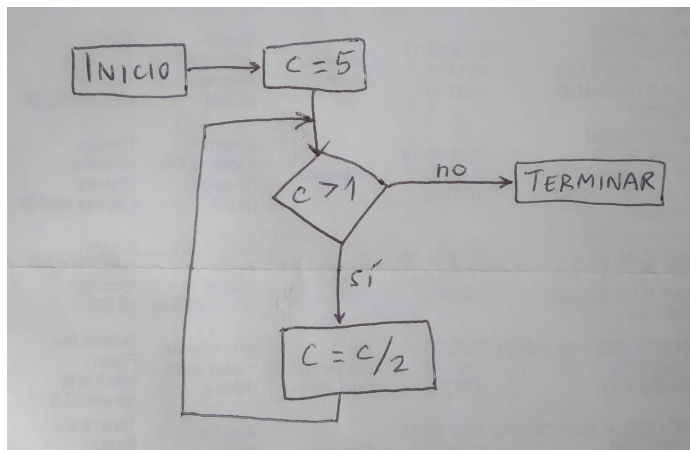
c	¿ c > 1 ?
5	sí
2.5	sí

1.25	sí
0.625	no

Puesto que al final ya no se cumple la condición, la computadora se detiene, quedando con el valor 0.625 en la variable «c».

Flujograma

El *flujograma* o diagrama de flujo es otra herramienta que le sirve al programador para entender los pasos que seguirá una computadora al ejecutar un algoritmo, pero esta vez de una forma gráfica. El flujograma del algoritmo anterior es:



[OJO: CORREGIR EN EL GUIÓN DEL VIDEO CORRESPONDIENTE O EN EL VIDEO EN SÍ]

Sentencia if

Otra sentencia de control, el 'if', se representa en un flujograma por medio de un rombo, de manera similar al del 'while' del flujograma anterior. Si la condición del 'if' se cumple, el programa *fluye* hacia un lado; si no se cumple, *fluirá* hacia otro lado.

La vida real está llena de sentencias *if*. Por ejemplo, supongamos esta ley tributaria: Si el sueldo de una persona es mayor que 1000 dólares, el impuesto a la renta que deberá pagar es del 4 %; caso contrario, solo pagará el 2 %. En un flujograma, esto sería:

(INCLUIR UN ROMBO donde en el interior dice “¿s>1000?”. Una flecha sale hacia la derecha, sobre ella aparece “sí” y desemboca en un rectángulo que dice “iR = 0.04*s”. Una flecha sale hacia abajo, a su izquierda aparece “no” y desemboca en un rectángulo que dice “iR = 0.02*s”)

Pseudocódigo

Dependiendo del lenguaje de programación, la forma en que se escriben las instrucciones puede cambiar un poco de lenguaje a lenguaje. A la forma en que se escriben las instrucciones en un determinado lenguaje (es decir, esa especie de normas gramaticales / ortográficas del lenguaje) se le denomina «sintaxis». Una sección de código que no tiene una sintaxis específica, sino que emplea incluso palabras del lenguaje natural humano, se denomina «pseudocódigo».

Algoritmos en el mundo actual

Vivimos en la llamada «era de la información», impulsada por el auge de la inteligencia artificial. Esta rama de la informática no es más que un conjunto de algoritmos que simulan la inteligencia biológica. Los algoritmos que hemos visto en este tema son mucho más simples que los de la IA, pero, aunque parezca raro, son la base y fundamento de ellos: en principio, todos los algoritmos se parecen, por muy simples o complejos que sean.

Así, es muy importante comprender los algoritmos sencillos y fundamentales... ¡El resto es añadir ingenio, creatividad e imaginación!

Actividades recomendadas

Realizar con los estudiantes una prueba de escritorio para el siguiente código. ¿Qué quedará almacenado en la variable «a» después de que este se ejecute?

```
a = 3
while a < 100
  a = a*a
(Respuesta: 6561)
```

Realizar con los estudiantes una prueba de escritorio para el siguiente código: ¿Qué quedará almacenado en la variable «a» después de que este se ejecute?

```
a = 3
for i = 1:4
  if i < 3
    a = a + i
  a = a + 1
(Respuesta: 10)
```