

Introducción al pseudocódigo





Pseudocódigo:

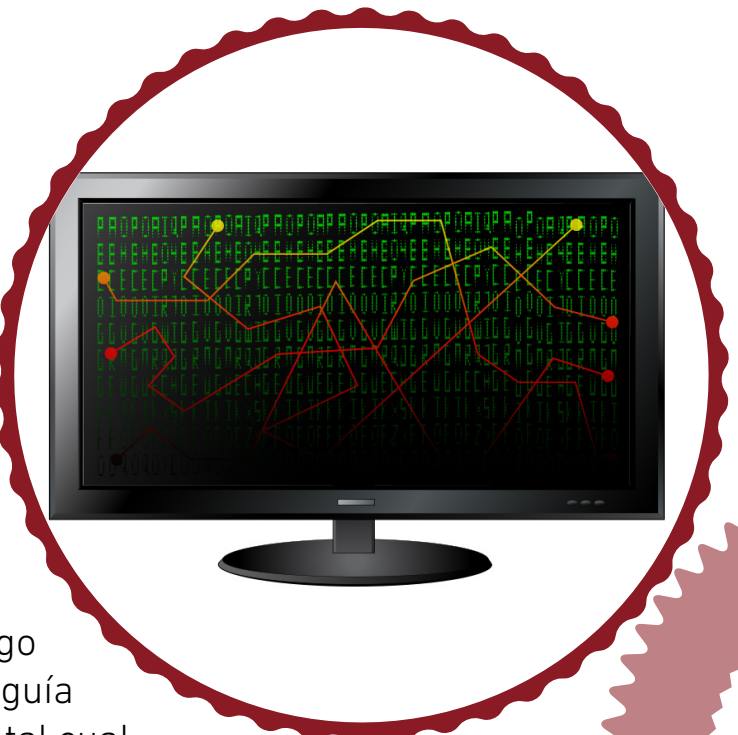
El pseudocódigo es un lenguaje de programación, el cual está basado en el lenguaje natural. Este tipo de programación permite definir la sintaxis o la estructura que llevará el programa de software a desarrollar. Allí, se estipularán las condiciones, repeticiones, variables a utilizar y las operaciones a realizar.

Una de las bondades de utilizar este método de programación para diseñar soluciones de software, es que permite corregir los errores lógicos y de sintaxis que tenga el algoritmo, para así corregirlos.



Palabras Reservadas:

Dentro de la estructura del pseudocódigo existen palabras, las cuales darán la guía para seguir el flujo de la información, tal cual se realizaba en los diagramas de flujo de datos. Estas palabras reservadas solo se utilizarán para indicar la operación o condición que se desea ejecutar. El siguiente es el listado de las palabras reservadas:



PALABRA	DESCRIPCIÓN
Inicio	Aquí se inicia el programa en pseudocódigo
Fin	Finaliza el proceso a realizar por el algoritmo.
Var	Se indican las variables a utilizar
Cadena	Variable tipo cadena de caracteres, por ejemplo "Hola Mundo"
Númérico	Indica que las variables son de tipo número
Carácter	Indica que la variable solo va a tener un símbolo, por ejemplo 'a'
Escribir	Impresión en pantalla de un mensaje para que el usuario lo pueda visualizar
Leer	Lee o guarda el valor que se le asigna a la variable
Si	Condicional
Entonces	Qué acción se debe ejecutar después de la condición
SiNo	Camino contrario a la condición inicial
Para	Ciclo de repetición
Mientras	Ciclo de repetición, se ejecuta mientras cumpla con una condición. Por ejemplo Mientras $a > 5$ entonces $a = a + 1$.
Repita hasta - que	Repetir hasta que se cumpla una condición final
Caso	Selección de opciones, lo que se utiliza para realizar menú de selección.
Subrutina	Nuevo programa para ejecutar
\n	Salto de línea
+	Concatenar dos cadenas de caracteres. Ejemplo "Hola"+"mundo"
Cls	Limpiar pantalla.
//	Comentarios de línea
/* * /*	Comentarios de bloque
<>	Asignar una cadena de caracteres a una variable, para generar las condiciones
Desde - hasta	Ciclo repetitivo que va desde una condición inicial hasta una condición final

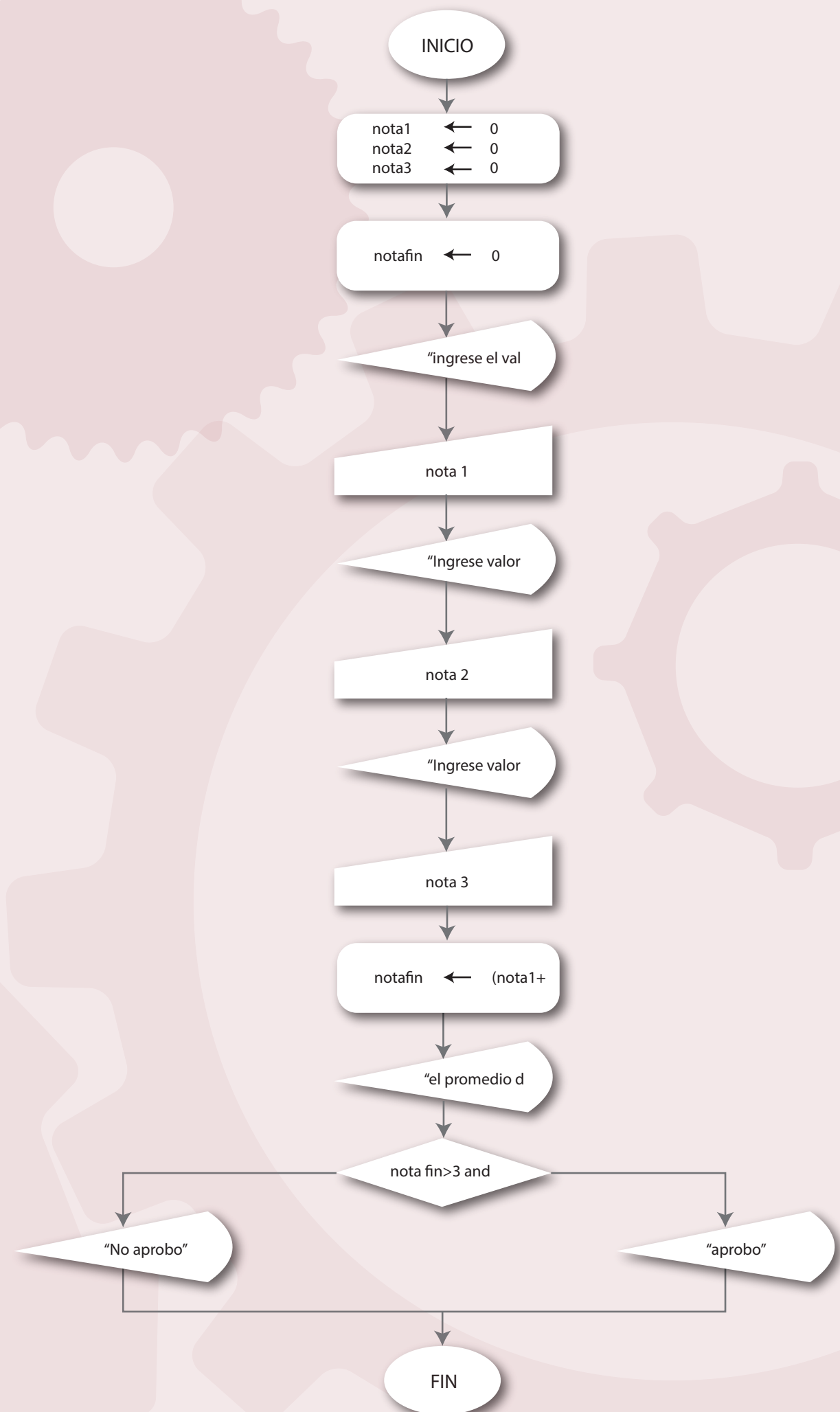
Luego de tener claro cada uno de las palabras reservadas, es necesario conocer los operadores aritméticos, ya que de ellos depende gran parte del buen funcionamiento de cada uno de los procesos que se realicen dentro del pseudocódigo.



Signo	Descripción
+	Suma
-	Resta
*	Multiplicación
/	División
Mod	Módulo de División
Div	División entera

Para ver el funcionamiento de cada uno de los anteriores operadores y palabras reservadas, a continuación se aplicará un ejemplo de la vida cotidiana. El problema que se presentará es el cálculo del promedio de notas de un estudiante.

Para poder solucionar la situación planteada, es necesario realizar el proceso desde el diseño del diagrama de flujo de datos y luego transformarlo en pseudocódigo.



Luego realizar el diseño del DFD y verificar la ejecución correcta del mismo, se procede a realizar el pseudocódigo:

```

Var nota1,nota2,nota3, notafin: numérico
inicio
  Escribir ("Ingrese el valor de la primera nota")
  Leer (nota1)
  Escribir ("Ingrese el valor de la segunda nota")
  Leer (nota2)
  Escribir ("Ingrese el valor de la tercera nota")
  Leer(nota3)
  notafin=(nota1+nota2+nota3)/3
  Escribir ("El Resultado de la suma es: ")
  Escribir(notafin)
  Si notafin=>3.0 AND notafin<=5.0 entonces
    Escribir ("Aprobó")
  SiNo
    Escribir ("No aprobó")
  FinSi
fin

```

En el ejemplo anterior se pudo detallar la utilización de las variables, de condicionales y de las diferentes palabras reservadas que contiene este lenguaje de programación.

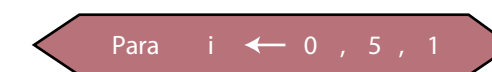
Luego de analizar el ejemplo anterior, con condicionales sencillas, es hora de conocer las estructuras repetitivas, las cuales ayudan a minimizar el tiempo de programación y de ejecución de un programa, ya que a partir de estas estructuras es más fácil realizar operaciones que se tengan que hacer un número de veces.

Entre las estructuras repetitivas se encuentran: para, el hacer – hasta, mientras, repetir, caso.

Estructura	Descripción
Para	Ciclo repetitivo que permite y ejecuta varias instrucciones, dependiendo de la condición de incremento o de decremento que el programador asigne, como también la asignación de valores a la variable de control del ciclo.
Mientras	Estructura repetitiva en aquellos cuya solución se utilice las mismas instrucciones, dependiendo del valor que se le asigne a la variable. Algunas veces, estas son variables lógicas que indican hasta dónde debe llegar el resultado final.
Repetir (Hacer – mientras)	Limita exactamente hasta dónde debe llegar el proceso, lo que quiere decir y evalúa que se cumpla la condición final. Ese proceso lo hace la instrucción Hacer. La instrucción Mientras evalúa si se cumple o no la expresión de verdad.

Aplicando cada una de las estructuras repetitivas dentro de la programación de pseudocódigo, un ejemplo sencillo para aplicarlos sería el de las notas, donde se ingresen seis notas, sin la necesidad de hacer los seis llamados de los datos. En diagrama de flujo de datos se observaría de esta forma:

El primer mensaje que podrá visualizar el usuario es "La cantidad de notas que desea promediar" y se guarda el valor ingresado en la variable cantn. El ciclo se representa en los DFD de la siguiente manera:



Esto traduce que para la variable de control, en este caso *i*, se le asignará un valor de cero, para así inicializar el proceso. Luego, sigue la sentencia que dará el límite hasta donde debe realizarse en ciclo. En este caso el valor es 5 y el incremento para que la función se realice de la mejor forma y cumpla con las normas dadas, se realizará en una unidad.

Luego de realizarse la sentencia para, se capturará el valor de la variable de las calificaciones, para luego realizar el cálculo del promedio. El dfd se observaría de la siguiente manera:

En pseudocódigo, utilizando en el ejemplo el ciclo para, se realizaría de la siguiente forma:

```

Var notafin, califi: numérico
Inicio
notafin=0
califi=0
Para i <-1 hasta 5 incremento en 1 hacer
Escribir ("Ingrese la "+ i +" Calificación")
Leer (califi)
notafin=notafin+calif
fin_para
Escribir ("El promedio es:")
Escribir (notafin/5)
fin

```

Ahora se observará qué pasa al utilizar un ciclo Mientras. Este ciclo es igual que realizar un condicional. Este se realiza con la evaluación de una expresión lógica; luego que se evalúe, cumple con la ejecución de una serie de instrucciones.

En pseudocódigo, la utilización de este ciclo se realiza de la siguiente manera:

```

Var nota1,nota2,nota3, notafin: numérico
Inicio
Escribir ("Ingrese el valor de la primera nota")
Leer (nota1)
Escribir ("Ingrese el valor de la segunda nota")
Leer (nota2)
Escribir ("Ingrese el valor de la tercera nota")
Leer(nota3)
mientras contador <5 AND notafin >= 3.0
notafin=(nota1+nota2+nota3)/3
fin_Mientras
Escribir (notafin)
Fin

```

Prueba de Escritorio:

La realización de la prueba de escritorio es la primera prueba funcional que se debe realizar en el diseño de un programa de software. Esta prueba consiste en mostrar los resultados, dependiendo de los caminos que tome en cada una de las operaciones necesarias.

Esta prueba de escritorio ayuda a encontrar errores existentes dentro del diseño del algoritmo, y poder corregir antes de la codificación final en un lenguaje de programación específico.

Para realizar esta prueba es necesario realizar una tabla donde se ubiquen cada una de las variables, las condiciones que se deben cumplir y la muestra los resultados.

Por ejemplo, para realizar la prueba de escritorio y del promedio de las notas, se colocarán las variables, el resultado y la toma de decisiones. Ahora, se mostrará el desarrollo de la prueba de escritorio. El primer paso es identificar las variables necesarias.

nota1, nota2, nota3, notafin

Luego identificamos las condiciones a realizar:

Si $\text{notafin} \geq 3.0$ AND $\text{notafin} \leq 5.0$ entonces

Y la operación a realizar:

$\text{notafin} = (\text{nota1} + \text{nota2} + \text{nota3}) / 3$

Ahora iniciamos la prueba de escritorio:

UNIDAD Virtual

Prueba de Escritorio (problema de notas)				
nota1	nota2	nota3	Notafin	$\text{notafin} \geq 3.0$ AND $\text{notafin} \leq 5.0$
3.0	3.0	4.0	$(3.0 + 3.0 + 4.0) / 3 = 3.33$	"Aprobo"
2.0	2.5	4.0	$(2.0 + 2.5 + 4.0) / 3 = 2.5$	"No Aprobo"

En el ejemplo anterior se realizó la prueba con diferentes valores, para ver las salidas generadas por cada uno de los caminos o decisiones tomadas.



SAN MATEO
Educación Superior

UNIDADVirtual