

Actividad | #1 | Algoritmos

Introducción al desarrollo de software

Ingeniería en Desarrollo de Software



TUTOR: Sandra Luz Lara Dévora

ALUMNO: Diego Barraza Carrillo

FECHA: 12 de agosto del 2024

Índice

INTRODUCCIÓN	3
DESCRIPCIÓN	3
JUSTIFICACIÓN	3
DESARROLLO.....	4
• CALCULADORA NÚMEROS PRIMOS	4
• CALCULADORA NUMEROS PAR E IMPAR	5
• CALCULADORA NÚMERO INVERTIDO	6
CONCLUSIÓN.....	8
REFERENCIAS.....	8

INTRODUCCIÓN

El pseudocódigo son una serie de palabras que se utilizan para describir los pasos que va a ejecutar un programa. Para ello se utilizan expresiones matemáticas, lógicas y variables.

El pseudocódigo es la representación escrita de un algoritmo, y un algoritmo es la representación de la solución de un problema eficientemente, por lo tanto, muestra los pasos a seguir para solucionar un problema de forma clara y sencilla. Su objetivo principal es interpretar las instrucciones de la resolución de un algoritmo, para luego ser utilizado con un lenguaje similar para ser codificado en un programa antes de escribir el código real, por lo que ayuda a planificar y entender el flujo del programa.

Los pseudocódigos pueden seguir diferentes estructuras, la primera, de manera secuencial en la que se ejecuta una instrucción tras otra y de arriba a abajo, o manera selectiva, en la que se ejecutan según una condición dada.

Utilizamos el pseudocódigo para crear un algoritmo finito, secuencial, preciso y lógico que cuenta con un inicio, un proceso y una salida en el que se muestra el resultado obtenido.

DESCRIPCIÓN

En la primera actividad integradora de la materia de Introducción al Desarrollo de Software se nos pide realizar varios algoritmos según un contexto presentado: Mathtech requiere un ingeniero en desarrollo de software para programar 3 calculadoras diferentes: La primera calculadora *Primos* tiene como objetivo identificar si los números ingresados son primos, es decir si es divisible entre 1 y si mismo, para luego arrojar “el número n es primo” o “el número n no es primo” según sea el caso.

La segunda calculadora *Par/impar* tiene como objetivo que se ingresen 10 números enteros y determinar e indicar cuáles números son pares o impares.

La última calculadora *Al revés* su objetivo es que el usuario ingrese un número de 4 dígitos y que sea un número entero, y este programa se encargará de regresar los número al revés. Por ejemplo, si se ingresa el número 7631, el programa matemático deberá regresar 1367.

JUSTIFICACIÓN

El objetivo y primer paso de esta actividad es comprender la funcionalidad de cada programa, y para ello debe escribirse en un algoritmo. Con la realización de estos algoritmos y uso de pseudocódigo se busca que desarrollemos nuestra habilidad analítica y lógica para planificar y diseñar programas que nos ayuden a resolver problemas específicos enfocados en las áreas de programación, de manera que nos permita optimizar el tiempo y los recursos. Nos facilita la comunicación de las ideas del algoritmo ya que el pseudocódigo no está escrito en un lenguaje de programación específico.

El pseudocódigo refuerza el identificar errores antes de perder tiempo con la codificación en un lenguaje

En el ámbito empresarial puede ayudarnos a crear programas que nos ayude a procesar datos, automatizar tareas para que se ejecuten en segundo plano mientras hacemos alguna otra actividad.

Básicamente se utiliza para solucionar un problema en relación con la programación.

DESARROLLO

- **CALCULADORA NÚMEROS PRIMOS**

1. Definimos las variables que vamos a utilizar: **num**, **x** y **contador** para números enteros y **opción** como caracter.

```

1  Algoritmo NumerosPrimos
2      Definir num, x, contador Como Entero
3      Definir opcion Como Caracter
4

```

2. Iniciamos con la indicación “repetir” sin cerrarla y escribimos la indicación “teclea un numero” para que aparezca en pantalla.

```

5      Repetir
6          Escribir "Teclea un numero"

```

3. e asignamos la variable **num** a los números que se van a teclear y el valor **1** a **x** y el valor 0 a **contador**.

```

7          leer num
8          x = 1
9          contador = 0

```

4. Utilizamos un ciclo **Mientras** para evaluar la variable **x** es menor o igual que **num**.

```

10     Mientras x ≤ num Hacer

```

5. En el bucle usamos la función **Mod** para evaluar si el residuo de **num** y **x** es una división exacta y con la variable **contador** contamos esas divisiones.

```

11         si num mod x == 0 Entonces
12             contador = contador + 1
13         FinSi

```

6. incrementamos la variable **x** para que retorne el bucle para que **x** sea igual a **num** y finalizamos el ciclo **Mientras**.

```

14             x = x + 1
15     FinMientras

```

7. Evaluamos si la variable **contador** es igual a 2. Si la función se cumple escribimos el número **num** es primo, sino se cumple escribimos número **num** no es primo.

```

17     si contador == 2 Entonces
18         Escribir "El numero ", num, " tecleado es primo"
19     SiNo
20         Escribir "El numero ", num, " tecleado no es primo"
21     FinSi

```

8. Después empleamos **repetir** y preguntamos si desea escribir otro número.

```

23     Repetir
24         Escribir "¿Deseas verificar otro número? (si/no)"

```

9. Leemos el carácter **opción**, si la opción es **si**, se repetirá el algoritmo, si la opción es **no**, terminará el algoritmo.

```

25         leer opcion
26         Hasta Que opcion == "si" O opcion == "no"
27         Hasta Que opcion == "no"
28 FinAlgoritmo

```

10. Finalizamos el pseudocódigo.

CODIGO COMPLETO:

```

1  Algoritmo NumerosPrimos
2  Definir num, x, contador Como Entero
3  Definir opcion Como Caracter
4
5  Repetir
6      Escribir "Teclea un numero"
7      leer num
8      x = 1
9      contador = 0
10     Mientras x ≤ num Hacer
11         si num mod x == 0 Entonces
12             contador = contador + 1
13         FinSi
14         x = x + 1
15     FinMientras
16
17     si contador == 2 Entonces
18         Escribir "El numero ", num, " tecleado es primo"
19     SiNo
20         Escribir "El numero ", num, " tecleado no es primo"
21     FinSi
22
23     Repetir
24         Escribir "¿Deseas verificar otro número? (si/no)"
25         leer opcion
26         Hasta Que opcion == "si" O opcion == "no"
27         Hasta Que opcion == "no"
28 FinAlgoritmo

```

• CALCULADORA NUMEROS PAR E IMPAR

- Empezamos dando valor 0 a **n** para que guarde nuestras iteraciones, y creamos la condición de nuestro ciclo, en este caso terminar la acción hasta haber indicado 10 números.

```

1
2   n = 0
3   Mientras n ≠ 10

```

2. Escribimos la indicación para que el usuario indique el número, y la línea que leerá dicho número.

```

4       Escribir "por favor digite un numero"
5       Leer numero

```

3. Utilizamos **mod** para dividir el número escrito sobre 2 y si resultado es 0 escribiremos que es número par, sino es impar.

```

6       si numero mod 2 = 0 Entonces
7           Escribir "el número " numero, " es par"
8       SiNo
9           Escribir "el número " numero, " no es par"
10      FinSi

```

4. Escribimos **n = n + 1** para que nuestros datos guardados se vayan sumando, hasta terminar con 10 datos ingresados. Y terminamos la condición y el algoritmo.

```

11      n = n + 1
12      FinMientras
13      FinAlgoritmo

```

CÓDIGO COMPLETO:

```

1  Algoritmo Numeros_par_e_impar
2      n = 0
3      Mientras n ≠ 10
4          Escribir "por favor digite un numero"
5          Leer numero
6          si numero mod 2 = 0 Entonces
7              Escribir "el número " numero, " es par"
8          SiNo
9              Escribir "el número " numero, " no es par"
10         FinSi
11         n = n + 1
12     FinMientras
13 FinAlgoritmo

```

• CALCULADORA NÚMERO INVERTIDO

1. Declaramos nuestras variables: **número** para guardar el número tecleado; **dígito** para cada dígito e **inverso** para guardar el número invertido.

```

1  Algoritmo Numero_invertido
2      numero = 0
3      digito = 0
4      inverso = 0

```

2. Damos la instrucción al usuario para que escriba el número y lo leemos.

```
5  Escribir "Escribe un número"
6  leer numero
```

3. Creamos la condición del bucle en la que el valor de **número** sea diferente a 0.

```
7  Mientras numero ≠ 0 Hacer
```

4. Escribimos **digito=numero%10**, para dividir sobre 10 el número original y guardar cada dígito en la variable **digito**.

```
8  digito=numero%10;
```

5. Utilizamos la función **trunc** para que la división nos dé números enteros.

```
9  numero=trunc(numero/10);
```

6. Usamos **inverso=inverso*10+digito** para el cálculo y obtener el número invertido.

```
10 inverso=inverso*10+digito;
```

7. Terminamos el bucle, mostramos en pantalla el número invertido y finalizamos el algoritmo.

```
11 FinMientras
12 Escribir "el número invertido es ", inverso;
13 FinAlgoritmo
```

CODIGO COMPLETO:

```
1  Algoritmo Numero_invertido
2  numero = 0
3  digito = 0
4  inverso = 0
5  Escribir "Escribe un número"
6  leer numero
7  Mientras numero ≠ 0 Hacer
8  digito=numero%10;
9  numero=trunc(numero/10);
10 inverso=inverso*10+digito;
11 FinMientras
12 Escribir "el número invertido es ", inverso;
13 FinAlgoritmo
```

CONCLUSIÓN

Para comprender cómo funcionan los programas es fundamental diseñar un algoritmo que describa sus pasos lógicos. La creación de estos algoritmos con un inicio, un proceso y un final, utilizan un pseudocódigo que actúa de manera secuencial o selectiva son utilizados de forma analítica y lógica para la planificación y diseño de programas que dan solución a distintos problemas en cuestión a programación. Este enfoque no solo nos ayuda a optimizar los procesos y los recursos que estos necesitan, sino también la comunicación entre los involucrados ya que el pseudocódigo se presenta en un lenguaje accesible, sin una sintaxis específica. Además, el uso de este lenguaje es importante para evitar pérdidas de tiempo antes de la fase de codificación, por lo que eludir los errores y e implementar posibles mejoras en la automatización de tareas, la mejora de eficiencia operativa y el procesamiento de datos es más segura y efectiva.

REFERENCIAS

Códigos en GitHub

<https://github.com/DiegoBarrazaC/Codigos-calculadoras>

Gálvez, J. A. S. (s. f.). *Pseudocódigo*. Unidades de Apoyo Para el Aprendizaje - CUAIEED - UNAM.

<https://uapa.cuaieed.unam.mx/sites/default/files/minisite/static/dd74ef5d-b946-4592-9b6c-139ecb42150e/UAPA-pseudocodigo/index.html>

TRUNCAR (TRUNC) - ayuda de editores de documentos de Google. (s. f.).

<https://support.google.com/docs/answer/3093588?hl=es-419>

Bucles en PSeInt: Automatizando Tareas Repetitivas / Blog Coders Free. (s. f.). Coders Free.

<https://codersfree.com/posts/bucles-en-pseint-automatizando-tareas-repetitivas>

De Roer, D. D., & De Roer, D. D. (2023, 25 julio). Operadores pseudocodigo. *Disco Duro de Roer* -.

<https://www.discoduroderoer.es/operadores-pseudocodigo/>