Universidad Corporativa Coppel | Programas Académicos

Ingeniería en Desarrollo de Software

Actividad | # 2 | Diagramas de flujo

Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software

academia global

TUTOR: Sandra Luz Lara Dévora

ALUMNO: Genaro Kantun Can

FECHA: 13-04-2025

Índice

Introducción……. 3

Descripción……...4

Justificación……..5

Desarrollo…….6

Diagrama de flujo número primo……. 6

Explicación diagrama de flujo número primo……...8

Diagrama de flujo par/impar…….9

Explicación diagrama de flujo par/impar……..11

Diagrama de flujo al revés……12

Explicación diagrama de flujo al revés…….14

Conclusión……16

Referencias……17

Link GitHub……..17

Introducción

¿Qué es un diagrama de flujo? Un diagrama de flujo es un esquema que representa un proceso o un procedimiento, indicando todos sus pasos, tareas o etapas de forma secuencial., los cuales cuentan con dos elementos fundamentales: símbolos y flechas o líneas; los diagramas de flujo sirven para organizar, evaluar o replantear secuencias de actividades.

Teniendo en cuenta este concepto, se procede a realizar los diagramas de flujo de los ejercicios de la actividad 1, en donde como ya se ha mencionado, se pretende realizar 3 calculadoras para apoyar a colegios y escuelas públicas, para realizar estos diagramas de flujo, ya se tiene que contar con los algoritmos realizados.

Para comprender bien los algoritmos, es importante tener en cuenta que constan de una serie de características, son sintéticos, porque al emplear símbolos y flechas, muestran de forma resumida todas las tareas de un procedimiento. Son simples porque se pueden elaborar rápidamente y son fáciles de comprender. Y son versátiles, porque pueden representar muchísimos procesos de diversas áreas.

Descripción

En informática y programación, los diagramas de flujo se utilizan para representar algoritmos, es decir, procesos que tienen una seria de pasos a seguir y que se emplean para diseñar el funcionamiento de programas, sitios web y aplicaciones.

Es por eso que en la presente actividad se estará realizando los diagramas de flujo de los ejercicios de la actividad 1, como se ha estado mencionando los diagramas de flujo son en base a los algoritmos que se emplearon para la realización de tres calculadoras, que servirán de apoyo a colegios y escuelas públicas.

Para poder realizar los diagramas de flujo, es muy importante tener en cuenta que ya se tenga terminado los algoritmos correspondientes a cada una de las calculadoras, tener en cuenta de igual forma, los símbolos correspondientes a cada acción que se va a ir realizando, en este sentido, y más allá de los orígenes ligados a la informática, esta herramienta permite tener una mayor organización.

Justificación

En esta actividad podremos observar la función principal de los diagramas de flujo, cada una de las partes que lo conforman, recordemos que para poder realizar un diagrama de flujo es importante tener realizado nuestros algoritmos, ya que estos son la base principal para la realización de los mismos.

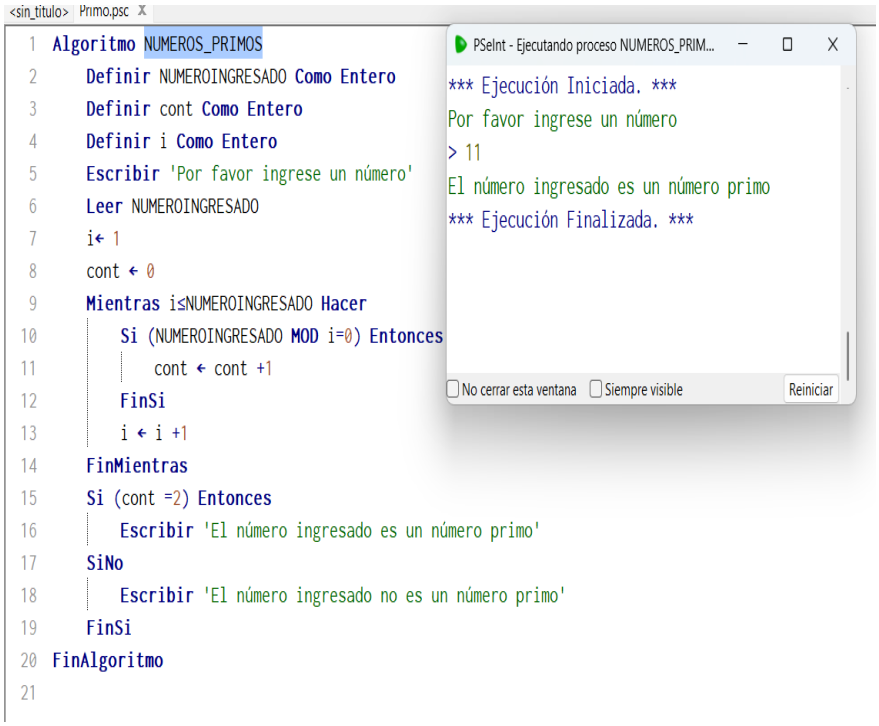
Los diagramas de flujo son una herramienta esencial para la mejora de la eficiencia, la productividad, y es aplicable en diversas áreas, desde la industria, la ingeniería hasta la educación, como es el caso de esta actividad, en donde los diagramas de flujo están destinados en representar los algoritmos de 3 calculadoras, cada una con una función y un tipo específico.

Recordemos que los algoritmos están conformados por formas y símbolos, en donde cada uno representa una función o una acción diferente, la capacidad que tienen para representar procesos de manera clara, convierte a los algoritmos en una herramienta indispensable para aquellos que busquen mejorar sus procesos y por consiguiente obtener buenos resultados.

Desarrollo

Diagrama de flujo Números Primos

```
Algoritmo NUMEROS_PRIMOS
    Definir NUMEROINGRESADO Como Entero
    Definir cont Como Entero
    Definir i Como Entero
    Escribir 'Por favor ingrese un número'
    Leer NUMEROINGRESADO
    i← 1
    cont ← 0
    Mientras i≤NUMEROINGRESADO Hacer
        Si (NUMEROINGRESADO MOD i=0) Entonces
            cont ← cont +1
        FinSi
        i ← i +1
    FinMientras
    Si (cont =2) Entonces
        Escribir 'El número ingresado es un número primo'
    SiNo
        Escribir 'El número ingresado no es un número primo'
    FinSi
FinAlgoritmo
```



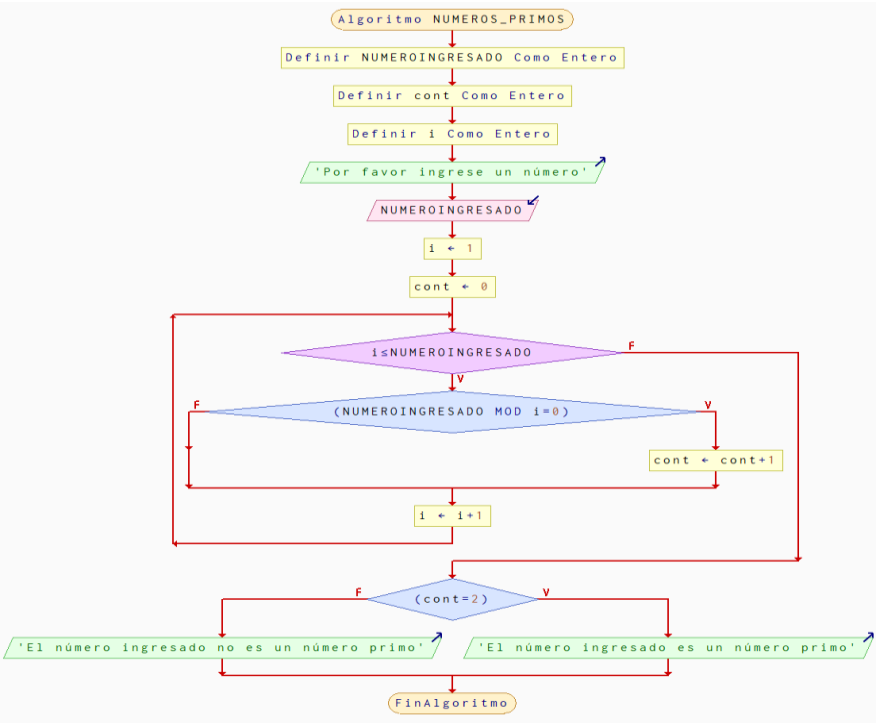
<sin_titulo> Primo.psc

```
1  Algoritmo NUMEROS_PRIMOS
2      Definir NUMEROINGRESADO Como Entero
3      Definir cont Como Entero
4      Definir i Como Entero
5      Escribir 'Por favor ingrese un número'
6      Leer NUMEROINGRESADO
7      i← 1
8      cont ← 0
9      Mientras i≤NUMEROINGRESADO Hacer
10         Si (NUMEROINGRESADO MOD i=0) Entonces
11             cont ← cont +1
12         FinSi
13         i ← i +1
14     FinMientras
15     Si (cont =2) Entonces
16         Escribir 'El número ingresado es un número primo'
17     SiNo
18         Escribir 'El número ingresado no es un número primo'
19     FinSi
20 FinAlgoritmo
21
```

PSeInt - Ejecutando proceso NUMEROS_PRIM...

```
*** Ejecución Iniciada. ***
Por favor ingrese un número
> 9
El número ingresado no es un número primo
*** Ejecución Finalizada. ***
```

No cerrar esta ventana Siempre visible Reiniciar



Explicación diagrama de flujo número primo

En el primer diagrama tenemos como título, número primo, en donde el primer paso es definir nuestras variables:

- Númeroingresado es el número que se va a evaluar, se toma como un número entero
- Cont, es un contador, es utilizado para saber cuántas veces se cumple una condición, de igual forma como un número entero.
- i representa el residuo de la división, también es un número entero.

Después de definir nuestras variables, se procede a darle un valor a nuestra variable Númeroingresado, esto para poder hacer la evaluación.

Posteriormente se agrega una serie de condiciones, acompañado de una operación relacional, en este caso "≤" (menor o igual que), estas condiciones se tienen que cumplir para que la función del algoritmo sea la correcta.

De igual forma se hace uso de la función matemática "Mod", el cual nos devolverá el residuo de la división.

Así mismo se agrega la condicional "Si" en donde se evalúa que nuestra condición sea igual a dos, si la condición se cumple mandará el siguiente mensaje "El número ingresado es un número primo", si la condición no se cumple, entonces mandará el siguiente mensaje "El número ingresado no es un número primo".

Y una vez cumplida las condiciones se da fin al algoritmo.

Diagrama de flujo Par/Impar

Par_impar.psc

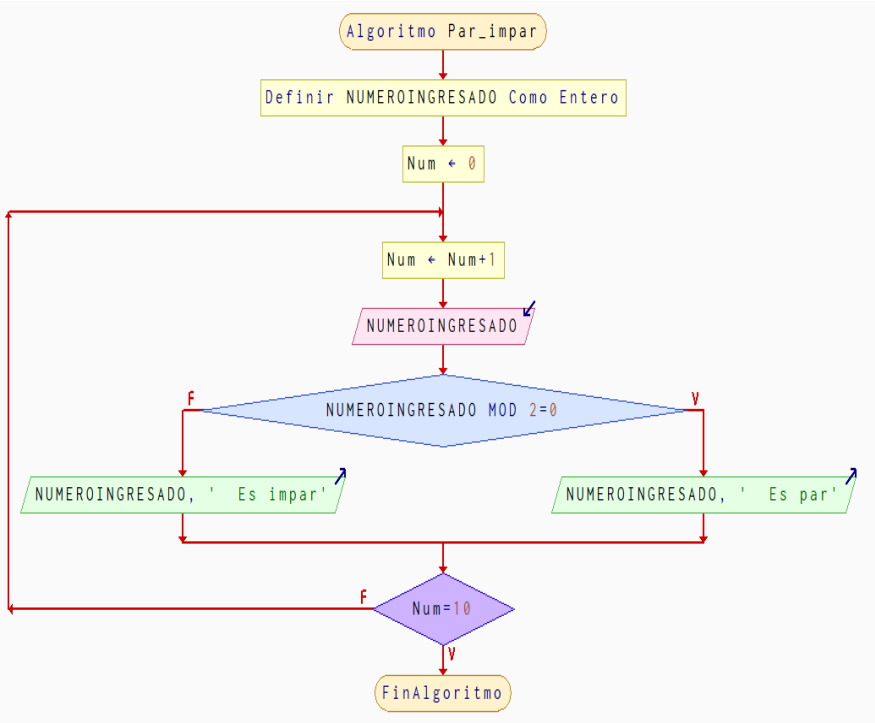
```
1  Algoritmo Par_impar
2      Definir Númeroingresado Como Entero
3      Num = 0
4      Repetir
5          Num = num + 1
6          Leer Númeroingresado
7          si Númeroingresado % 2 =0 Entonces
8              Escribir Númeroingresado, "  Es par"
9          SiNo
10             Escribir Númeroingresado, "  Es impar"
11         FinSi
12     Hasta Que Num = 10
13 FinAlgoritmo
14
```

PSeInt - Ejecutando proceso PAR_IMPAR

```
*** Ejecución Iniciada. ***
por favor ingrese un número
> 2
2  Es par
por favor ingrese un número
> 14
14  Es par
por favor ingrese un número
> 100
100  Es par
por favor ingrese un número
> 40
40  Es par
por favor ingrese un número
> 6
6  Es par
por favor ingrese un número
> 9
9  Es impar
```

```
por favor ingrese un número
> 11
11  Es impar
por favor ingrese un número
> 101
101  Es impar
por favor ingrese un número
> 17
17  Es impar
por favor ingrese un número
> 21
21  Es impar
*** Ejecución Finalizada. **
```

☐ No cerrar esta ventana ☐ Siempre visible



Explicación diagrama de flujo par/impar

Nuestro segundo diagrama tiene como título par/impar, definimos variables, en este caso “Númeroingresado” que será como entero.

Num, primeramente, se le asigna el valor “0” ya que desde ahí se va a empezar a contar hasta que lleguemos a contar 10 números.

Seguidamente agregamos el comando “repetir”, que se va a encargar de repetir el ciclo de nuestra condición, a “Num” le asignamos el valor de 1 para que nos vaya contando los números ingresados.

Leemos “Númeroingresado”, iniciamos nuestra condición “Si” en donde le decimos al programa que el “Númeroingresado” se va a dividir entre 2 y si el resultado es “0”, entonces le decimos al programa que imprima el mensaje “es par” si la condición no se cumple entonces el mensaje sería el siguiente “es impar” así hasta que lleguemos a un conteo total de 10 números.

Una vez llegado al conteo total, se da fin al algoritmo.

Diagrama de flujo Al revés

Al revés.psc

```
Algoritmo Al_revés
    definir n, b Como Entero
    definir a, x Como Caracter
    Escribir "Por favor ingresa un número de 4 cifras"
    leer n
    a = ConvertirATexto(n)
    b = longitud (a)
    x = ""
    si n ≥ 1000 y n ≤ 10000 Entonces
        mientras b > 0 hacer
            x = x + subcadena (a, b , b)
            b = b - 1
        FinMientras
        Escribir "El número ",n," invertido es: ", ConvertirANumero(x)
    sino
        Escribir "Por favor vuelva a intentarlo"
    FinSi
FinAlgoritmo
```

Al revés.psc

```
Algoritmo Al_revés
    definir n, b Como Entero
    definir a, x Como Caracter
    Escribir "Por favor ingresa un número de 4 cifras"
    leer n
    a = ConvertirATexto(n)
    b = longitud (a)
    x = ""
    si n ≥ 1000 y n ≤ 10000 Entonces
        mientras b > 0 hacer
            x = x + subcadena (a, b , b)
            b = b - 1
        FinMientras
        Escribir "El número ",n," invertido es: ", ConvertirANumero(x)
    sino
        Escribir "Por favor vuelva a intentarlo"
    FinSi
FinAlgoritmo
```

PSeInt - Ejecutando proceso AL_REVES

```
*** Ejecución Iniciada. ***
Por favor ingresa un número de 4 cifras
> 5698
El número 5698 invertido es: 8965
*** Ejecución Finalizada. ***
```

No cerrar esta ventana Siempre visible Reiniciar

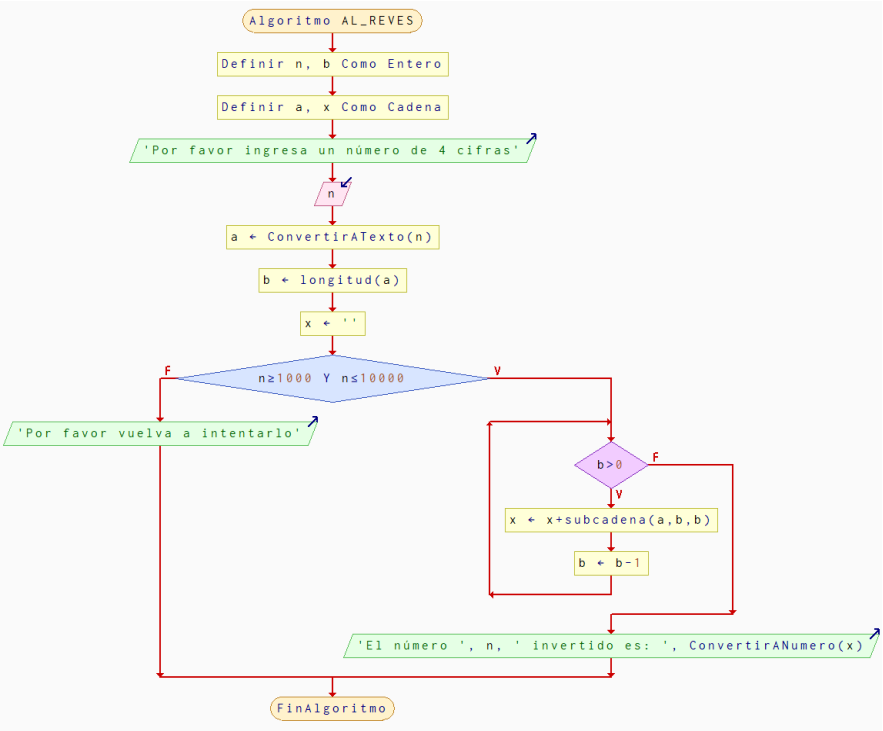
Al revés.psc

```
1  Algoritmo Al_revés
2      definir n, b Como Entero
3      definir a, x Como Caracter
4      Escribir "Por favor ingresa un número de 4 cifras"
5      leer n
6      a = ConvertirATexto(n)
7      b = longitud (a)
8      x = ""
9      si n ≥ 1000 y n ≤ 10000 Entonces
10         mientras b > 0 hacer
11             x = x + subcadena (a, b , b)
12             b = b - 1
13         FinMientras
14         Escribir "El número ",n," invertido es: ", ConvertirANumero(x)
15     sino
16         Escribir "Por favor vuelva a intentarlo"
17     FinSi
18 FinAlgoritmo
```

PSeInt - Ejecutando proceso AL_REVES

```
*** Ejecución Iniciada. ***
Por favor ingresa un número de 4 cifras
> 12255
Por favor vuelva a intentarlo
*** Ejecución Finalizada. ***
```

No cerrar esta ventana | Siempre visible | Reiniciar



Explicación diagrama de flujo al revés

Como nuestro último diagrama tenemos al que pondremos como título al revés.

Primeramente, definimos nuestras variables, en este caso la variable “n” y la variable “b”, las declaramos de tipo entero, de igual forma agregamos las variables “a” y “x” las declaramos de tipo carácter.

La función “escribir” agregamos “Por favor ingresa un número de 4 cifras”, leemos el número (n).

Ya que tenemos el número, lo vamos a convertir a texto. Es decir “a” le asignamos la función “convertiratexto” y agregamos el valor de “n”.

Posteriormente decimos que “b” es igual a y utilizamos la función “longitud” y le agregamos el valor de “a”, en donde vamos a decir el tamaño que tiene “n”.

Seguidamente dejamos a “x” con un espacio en blanco.

Iniciamos con una condición con “si-entonces” en donde declaramos que si “n” es ≥ a 1000 y agregamos un operador lógico “y”, “n” es ≤ 10000, esto es para hacer que el programa reconozca que ingresamos un número de 4 cifras. Si la condición se cumple el programa iniciará con el ciclo,

Si la condición no se cumple el programa imprimirá el siguiente mensaje “por favor vuelva a intentarlo”.

Posteriormente iniciamos un ciclo en donde vamos a evaluar que “mientras” la variable “b” que es donde guardamos la longitud del número, mientras sea valor a “0”, vamos a “hacer” lo siguiente:

Declaramos que “x” es igual a lo que tiene la variable “x” (recordemos que utilizamos un espacio en blanco) le vamos a añadir un operador matemático “+” y la función “subcadena” y vamos a tomar las variables “a, b, b”, con esto estamos haciendo es que el programa tome el último

carácter del número y lo estamos almacenado en la variable “x”.

Después vamos a decir que la variable “b” es igual a lo que tenga “b” menos “1”, es decir, cuando el ciclo retorne vamos a tomar el penúltimo carácter, esto se va a hacer con todos los caracteres, por lo tanto, al finalizar el ciclo, el valor invertido se almacenará en la variable “x”, al terminar el ciclo, si la primera condición se cumple, el programa imprimirá el siguiente mensaje “El número”, “n”, “invertido es”, pero recordemos que el valor de “x” es texto, entonces en este caso vamos a utilizar la función “convertiranumero”, para que el valor nos lo muestre en números.

Y con esto le damos fin al algoritmo.

Conclusión

Al realizar la presente actividad, aprendimos la utilización de diagramas de flujo, al igual de los elementos que los conforman, haciendo de esta herramienta muy útil en cualquier área laboral y en nuestra vida cotidiana, mejorando la eficiencia y la productividad mejorando la solución de problemas.

En cada elemento que lo conforman pudimos notar que cada uno tiene una función específica para poder darnos una solución de manera fácil y sencilla, esto nos ayudará, ya que una la base de toda creación de cualquier aplicación.

Así como tiene elementos, también hay varios tipos de diagramas de flujo, al utilizarlos podemos simplificar procesos y esto nos ayuda a disminuir errores que podrían hacer que el algoritmo no funcione de manera correcta.

En resumen, podemos decir que los diagramas de flujo en conjunto con los algoritmos, (ya que uno es la base de la creación del otro), son una herramienta que nos permitirán la mejora de la eficiencia, la productividad e innovación de diversas áreas.

Referencias

Concepto (s.f) Diagrama de flujo- ¿Qué es?, tipos, simbologías y ejemplos. Conceptos. Consultado el 2 de abril de 2025. https://concepto.de/diagrama-de-flujo/

Link Github

https://github.com/genkc/Introducci-n-al-desarrollo-de-software-