



27 DE JULIO DEL 2025

ACTIVIDAD # 2 DIAGRAMAS DE FLUJO

INTRODUCCION AL DESARROLLO DE SOFTWARE

Ingeniería en desarrollo de Software



ALUMNO: GERMAN NAHUAT NAHUAT



Índice

● Introducción -----	3
● Descripción -----	4
● Justificación -----	5
● Desarrollo -----	6
● Conclusión -----	12
● Referencias	

Introducción

Los algoritmos como hemos visto en la actividad anterior es una secuencia de pasos de manera ordenada de nuestras actividades cotidianas tanto personal como en nuestros centros de trabajo, al igual se pueden representar de manera gráfica, este método lo conocemos como diagramas de flujo. Los diagramas de flujo son la representación grafica de un algoritmo, donde su objetivo principal es ordenar de la mejor forma los pasos a seguir de un algoritmo.

Un diagrama de flujo debe cumplir con ciertas características, los cuales se componen de bloques de información así como flechas donde indica el flujo del proceso, al igual que el algoritmo tiene un inicio y un fin, se simboliza para que tenga un mejor rendimiento, permite leer mas rápido un proceso y nos sirve como una herramienta de control para organizar información.

Gracias a los diagramas de flujo en la actualidad podemos transmitir información a diferentes personas, ya que facilitan su entendimiento, como hemos visto hay diferentes tipos de diagramas solo es cuestión de identificar cual es que mas se acopla a nuestra información o algoritmo que se quiere hacer.

Descripción

En esta actividad estaremos desarrollando una serie de pasos conocidos como algoritmos donde se estará solucionando cada problemática planteada en las siguientes problemas:

La empresa MathTech requiere a un ingeniero en desarrollo de software que sea capaz de realizar la tarea de programar tres tipos de calculadoras diferentes para implementar en los colegios y escuelas públicas:

- La primera calculadora deberá de llevar por nombre Primos, y su objetivo será identificar los números primos que se ingresen, por ejemplo, si el usuario ingresa el número 83, deberá imprimir el siguiente mensaje: “El número (número ingresado) si es primo”, en caso de que no sea primo se imprimirá el siguiente mensaje “El número (número ingresado) no es primo”. Básicamente se encargará de identificar si un número es divisible entre 1 y el mismo.
- La segunda calculadora se llamará Par/Impar, su objetivo es que se ingresen 10 números, ya sean pares o impares, por ejemplo, si se ingresa el número 9, el programa deberá de indicar que es un número impar, pero si se trata del número 2, el programa deberá indicar que se trata de un número par. De 10 números enteros, se debe determinar cuáles son pares y cuáles son impares.
- El último programa se llamará 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 números al revés o invertidos. Por ejemplo, si se ingresa el número 7631, el programa matemático deberá regresar 1367.

Dado los problemas siguientes se procederá a identificar paso a paso lo que se debe hacer para darle solución a cada uno, respetando lo que hemos aprendido previamente, así creamos un algoritmo para cada problema y su diagrama de flujo de cada uno

Justificación

Como hemos estado analizando los diagramas de flujo facilita la interpretación de los algoritmos, algunas normas establecidas para hacer un diagrama de flujo son identificar las ideas principales, definir los objetivos, identificar que tipo de diagrama de flujo se estaría usando y elegir los detalles, así como los límites que tendrá.

Con esto podemos determinar la interpretación de los diagramas, también recalcar que los diagramas tenemos diferentes modelos, como la vertical, la horizontal y el formato panorámico.

Para una mejor interpretación de los algoritmos realizados para la actividad solicitada, para poder leer e interpretar mejor estos algoritmos se anexará su diagrama de flujo de cada uno para que se vea desde el momento que inicia cada algoritmo y los pasos que sigue para poder emitir el resultado de cada uno, así la persona que vea esto tenga la noción de los pasos que sigue un algoritmo para definir un resultado.

Desarrollo

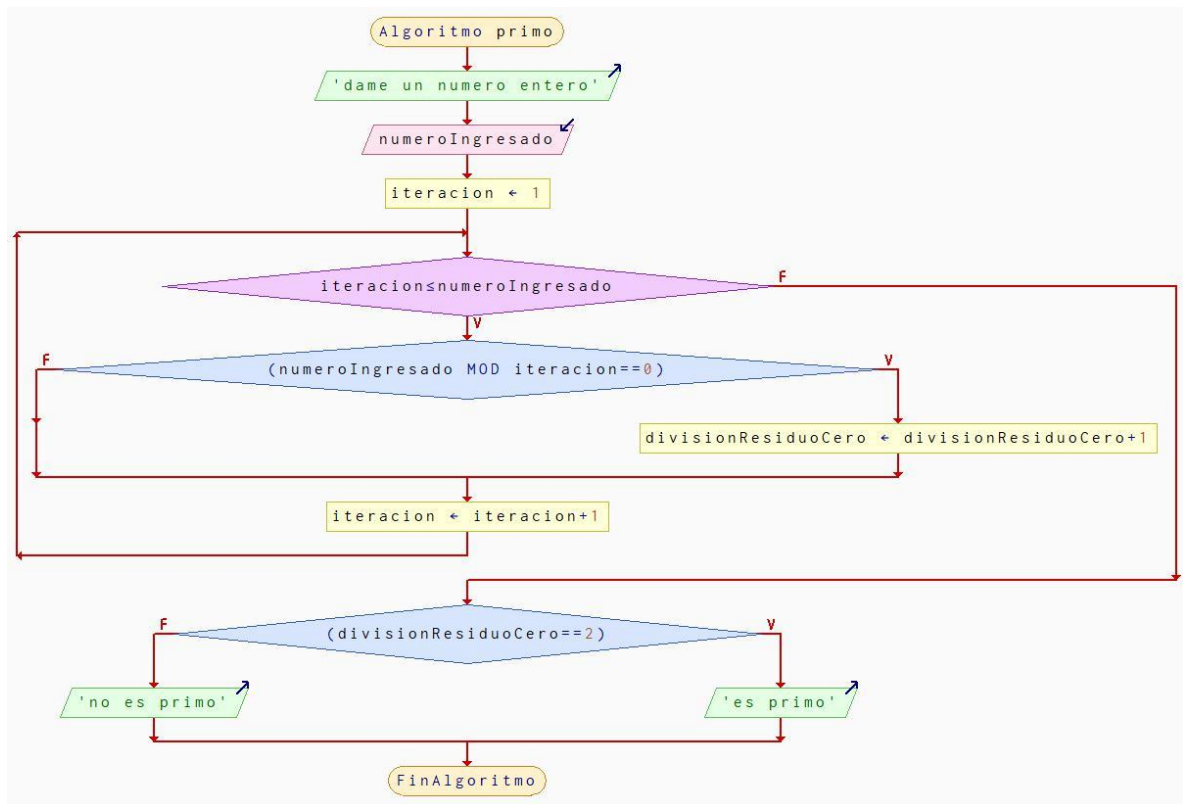
La primera calculadora deberá de llevar por nombre Primos, y su objetivo será identificar los números primos que se ingresen, por ejemplo, si el usuario ingresa el número 83, deberá imprimir el siguiente mensaje: “El número (número ingresado) si es primo”, en caso de que no sea primo se imprimirá el siguiente mensaje “El número (número ingresado) no es primo”. Básicamente se encargará de identificar si un número es divisible entre 1 y el mismo.

1. Definir algoritmo primo
2. Escribir número entero.
3. Leer número ingresado
4. Generar la viable de iteración.
5. Definir ciclo mientras.
6. Definir la condición de iteración.
7. Definir la división a realizar para determinar si es número primo
8. Si el residuo de la división es igual a 2 entonces es primo.
9. Si el residuo de la división es diferente de 2 entonces no es primo.
10. Fin del algoritmo

Veamos la secuencia de algoritmos ya ingresados en el programa que usamos.

- Algoritmo primo
- Escribir "dame un numero entero"
- Leer numeroIngresado;
- iteracion=1;
- Mientras iteracion <= numeroIngresado Hacer
- si(numeroIngresado%iteracion==0) Entonces
- divisionResiduoCero=divisionResiduoCero+1;
- FinSi
- iteracion=iteracion+1
- FinMientras
- si(divisionResiduoCero==2)Entonces
- Escribir "es primo"
- SiNo
- Escribir "no es primo"
- FinSi
- FinAlgoritmo

Este es la representación grafica de nuestro algoritmo para definir si un numero es Primo o No Primo.



En este diagrama se ve donde se ingresa el numero entero al igual en la secuencia se ve la iteración si es falso se pasa directamente a la división de residuo donde termina f si es NO PRIMO y verdadero si es PRIMO, pero si la iteración es verdadero entra al mod iteración donde el resultado ya sea falso o verdadero regresa de nuevo a la iteración hasta tener el resultado deseado.

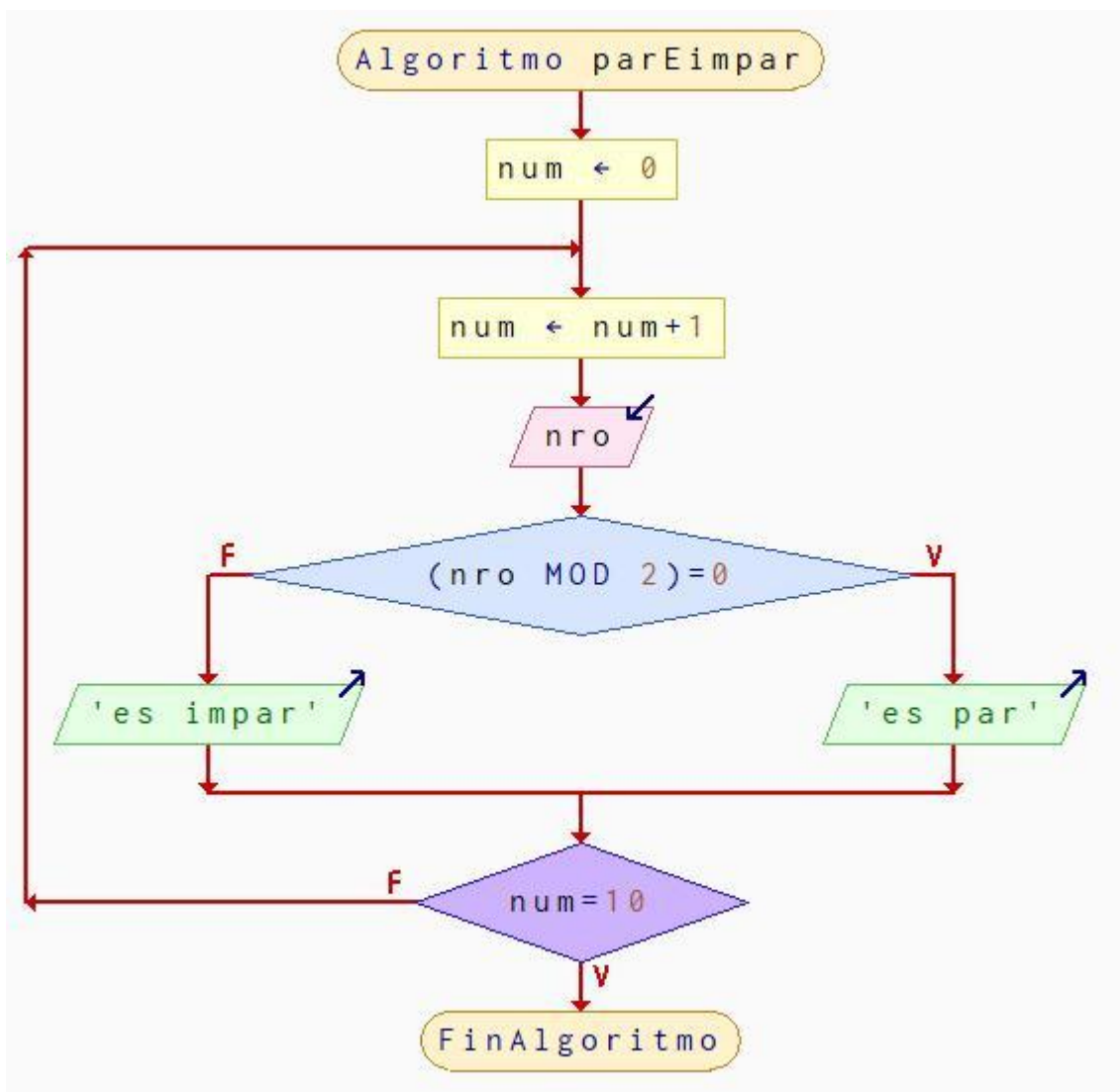
La segunda calculadora se llamará Par/Impar, su objetivo es que se ingresen 10 números, ya sean pares o impares, por ejemplo, si se ingresa el número 9, el programa deberá de indicar que es un número impar, pero si se trata del número 2, el programa deberá indicar que se trata de un número par. De 10 números enteros, se debe determinar cuáles son pares y cuáles son impares.

1. Definir algoritmo
2. Establecer la variable num para los números
3. Establecer el ciclo de repeticiones
4. Establecer las iteraciones del ciclo
5. Leer numero
6. Guardar una variable, ya que hará la función a cualquier numero
7. Establecer la condición.
8. Si es divisible entre 2 y el residuo o resultado es igual a cero(0), ese numero es par.
9. Si se divide entre 2 y el residuo no es igual o mayor a cero(0), ese numero es impar.
10. Fin de algoritmo.

Veamos la secuencia de algoritmos ya ingresados en el programa que usamos.

- Algoritmo parEimpar
- num=0;
- Repetir
- num=num+1;
- leer nro;
- si (nro mod 2)=0 Entonces
- Escribir "es par"
- SiNo
- Escribir "es impar"
- FinSi
- Hasta Que num=10
- FinAlgoritmo

Esta es la secuencia grafica de nuestro algoritmo para definir un numero Par e Impar, igual vemos toda la secuencia para que nos pueda dar el resultado solicitado.



En este diagrama de flujo podemos identificar el seguimiento del algoritmo, como la variable num y el ciclo definido al momento de ingresar el numero hace que el algoritmo define si es par o no es par un numero siguiendo si es falso de IMPAR, pero si es verdadero es PAR, al igual si sale un error vuelve a regresar al inicio para hacer el mismo procedimiento.

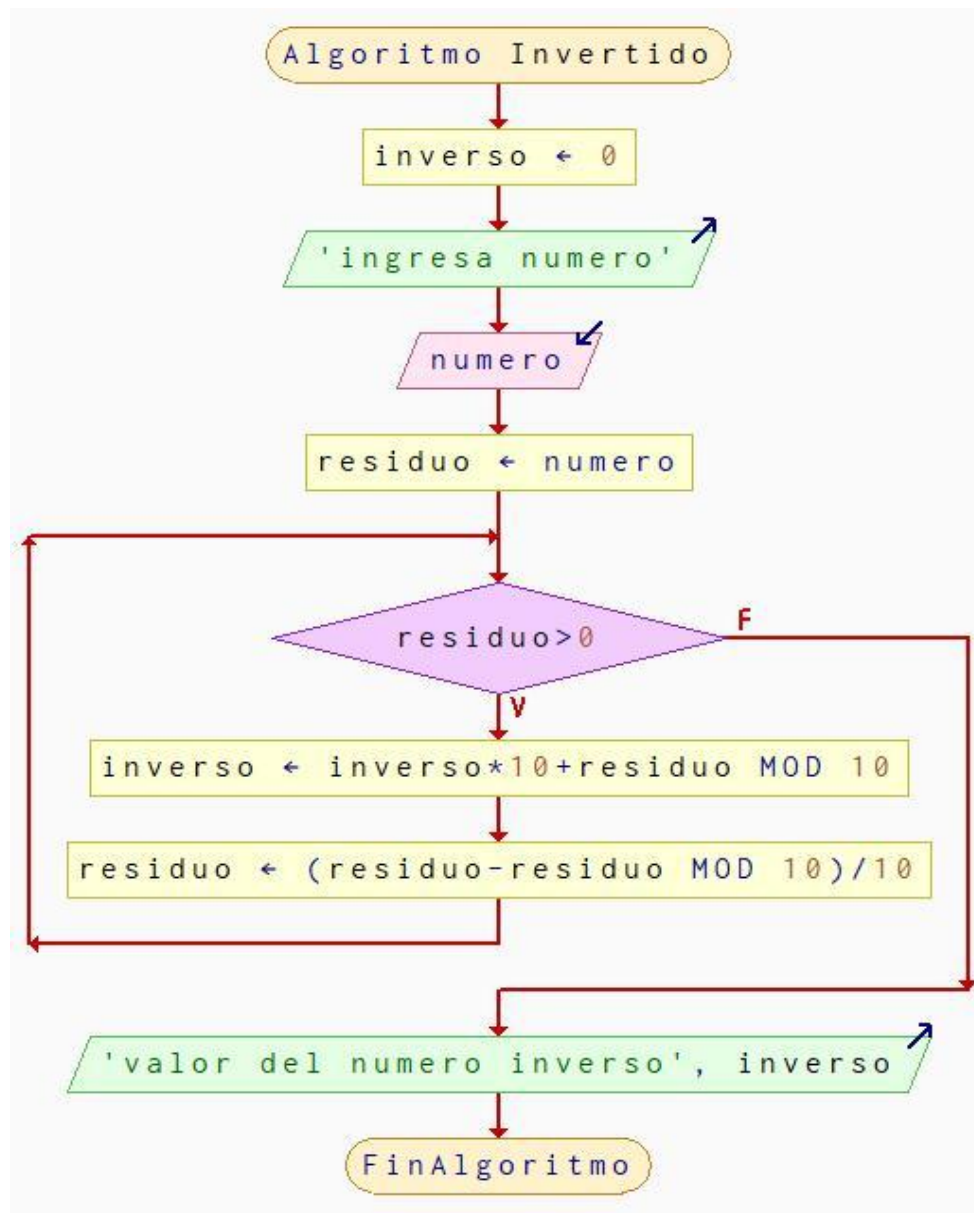
El último programa se llamará Al Revés, su objetivo es que el usuario ingrese un número de 4 dígitos y que sea un número entero, y este programa se encargará de regresar los números al revés o invertidos. Por ejemplo, si se ingresa el número 7631, el programa matemático deberá regresar 1367.

1. Definir algoritmo invertido
2. Asignación inversa a cero.
3. Escribir ingresar número.
4. Leer número.
5. Definimos secuencia.
6. Residuo del número mayor que cero.
7. Definimos ciclo
8. Ingresamos inverso por 10 más residuo mod 10
9. Residuo menos mod 10 dividido entre 10
10. Fin de ciclo
11. Escribir valor inverso
12. Fin del algoritmo.

Veamos la secuencia de algoritmos ya ingresados en el programa que usamos.

- Algoritmo Invertido
- `inverso <- 0`
- Escribir "ingresa numero"
- Leer numero
- `residuo <- numero`
- mientras `residuo > 0` Hacer
- `inverso <- inverso * 10 + residuo mod 10;`
- `residuo <- (residuo - residuo mod 10) / 10;`
- FinMientras
- Escribir "valor del numero inverso",inverso;
- FinAlgoritmo

Nuestro algoritmo de numeración invertido se muestra de la siguiente forma, contemplando paso a paso el algoritmo señalado para poder brindar el resultado programado.



En este algoritmo invertido vemos como al ingresar el número, desde el algoritmo realiza la división donde el residuo determina si es falso da el número inverso al solicitado, pero en caso de ser verdadero ingresa a las operaciones donde el residuo cumple con lo solicitado y regresa de nuevo para determinar el resultado y poder cumplir con el algoritmo.

Conclusión

En esta actividad fuimos viendo los diferentes diagramas de flujo y la utilización en la interpretación de un algoritmo, al igual que lo que se necesita para ir creando dependiendo del algoritmo o lo que se quiere representar, una de las herramientas para poder leer y comprender de manera fácil un algoritmo, igual aprendimos las partes que debe contener un diagrama de flujo, al igual que un algoritmo tiene un inicio y también tiene un fin, más bien representa todos los pasos que seguimos en un algoritmo de manera simplificada.

Los pasos para poder crear un diagrama tienen que contener las entradas y salidas de los procesos, ordenando así la información para poder plasmarlo eligiendo la simbología adecuada para su creación, también identificar los puntos de decisión para construir el diagrama respetando los lineamientos establecidos así poder probar un programa para que no exista errores.

Liga GITHUB

<https://github.com/German-Nahuat/Repositorio.git>