



21 DE JULIO DEL 2025

ACTIVIDAD # 1 ALGORITMOS

INTRODUCCION AL DESARROLLO DE SOFTWARE

Ingeniería en desarrollo de Software



ALUMNO: GERMAN NAHUAT NAHUAT



Índice

● Introducción -----	3
● Descripción -----	4
● Justificación -----	5
● Desarrollo -----	6
● Conclusión -----	9
● Referencias	

Introducción

El pseudocódigo está constituido por una serie de palabras que describen paso a paso sobre la ejecución de un programa, su principal objetivo es interpretar las instrucciones para la resolución de un algoritmo, como hemos estado aprendiendo en esta materia los algoritmos son finitos.

Los algoritmos tienen como objetivo principal resolver o dar solución a un problema de manera clara, para que pueda ser desarrollado en un lenguaje de programación. Como se ha leído anteriormente es finito, debe tener una secuencia, tiene que ser preciso al igual contar con una lógica, todo esto debemos considerarlo al momento de hacer los pasos de un algoritmo.

En esta actividad estaremos desarrollando un algoritmo siguiendo los pasos previamente explicados, al igual contemplando las siguientes partes. Debe contener una entrada donde se introduce la información para trabajar en ello, el proceso aquí describiremos los pasos a seguir para poder darle solución al problema y por último la salida, en esta parte final veremos los resultados finales de todo el proceso.

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.

Justificación

Hemos estado analizando sobre pasos para poder hacer algoritmos, hay que recordar que nuestra vida cotidiana también depende de algoritmos, al momento de levantarnos, los pasos que seguimos para poder hacer nuestras actividades cotidianas, de manera inconsciente lo hacemos para poder lograr un mismo objetivo, solucionando algún problema cotidiano, tales como solucionar el hambre, sed o parte de algún trabajo formal, por eso en esta actividad estaremos realizando algoritmos para poder solucionar los problemas que nos establecen, cada problema tiene una solución diferente para resolver, por lo que iremos catalogando cada uno, estableciendo cada algoritmo, hasta completar los 3 soluciones.

Siguiendo en cada uno los pasos previamente establecidos para poder completar las etapas del desarrollo de algoritmos establecidos, tenemos que tener la definición, análisis, diseño y de ultimo llevarlo a la implementación del algoritmo.

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 numero 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

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

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

Conclusión

En nuestra vida cotidiana hacemos varias cosas desde que nos levantamos tenemos una secuencia de tareas o acciones a realizar en el transcurso del día, todo esto se traduce en algoritmos, al igual en nuestra vida diaria usamos la calculadora sin saber el proceso que lleva hacer la programación de estas como lo hemos analizado en esta actividad. Ya sabemos que los algoritmos se hacen de manera ordenado siguiendo las parte que lo conforman al igual que las etapas para su desarrollo.

Recordemos que un algoritmo tiene que cumplir con las características mencionados anteriormente, todo lo que tiene un inicio también tiene un fin, todo eso se puede contemplar como algoritmo y todo esto se puede representar como un código al igual como ver la secuencia mediante su diagrama de flujo que estaremos explicando en actividades posteriores.

LIGA GITHUB

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