



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Alejandro Esteban Pimentel Alarcon

*Asignatura:* Fundamentos de programación

*Grupo:* 3

*No de Práctica(s):* 03

*Integrante(s):* Gómez Mendoza Jonan

*No. de Equipo de  
cómputo empleado:* 34 Perú

*No. de Lista o Brigada:*

*Semestre:* 2020-1

*Fecha de entrega:* 2/09/2019

*Observaciones* Excelente

**CALIFICACIÓN:** 10

# Introducción

Un algoritmo es un conjunto ordenado e inequívoco de pasos ejecutables que definen un proceso finito por ello siempre necesitan ser muy claros y expresados de manera literal, es necesario definir bien el algoritmo con precondiciones.

El ciclo de vida del software es el siguiente:



Como se puede apreciar constantemente se debe de revisar el algoritmo para optimizarlo

# Objetivo

Elaborar algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de análisis y diseño pertenecientes al ciclo de vida del software

## Actividades

Precondiciones y salidas para; Se facilita mucho si resaltas las subsecciones

Pescar: Las precondiciones son estar cerca de un lago, río donde existan peces, tener manos, una caña de pescar, carnada y la salida es un almacén con peces

Lavarse las manos: Las precondiciones son tener, manos, agua, jabón y la salida es manos limpias

Cambiar una llanta: Las precondiciones son tener un carro, una llave de cruz, un neumático nuevo, un gato hidráulico, y la salida es un carro con llanta cambiada

Convertir un numero binario a decimal: Las precondiciones son tener un número binario, saber contar, saber multiplicar, saber sumar y la salida es un número decimal

Se debe de destacar la importancia de las precondiciones ya que si no se cumplen algunas de ellas seria imposible realizar la tarea

Algoritmos para;

Determinar si un numero es positivo o negativo:

Precondiciones que el numero sea diferente de 0

Al número se le multiplica por si mismo luego se le saca raíz cuadrada positiva luego se le suma el numero original si es igual a 0 es un número negativo si no es igual a 0 es positivo

Terminar

Wow, muy original, te felicito por usar acciones que una máquina puede entender había formas mas sencillas, la computadora también sabe usar: < >

Obtener el mayor de dos números diferentes:

Precondiciones Debe de ser un número real y definido

Si un número es positivo y el otro es negativo siempre va a ser mayor el positivo

Restarle 1 al primer número si es igual al otro número el primer número es mayor en caso contrario verificar si es igual a 0 si lo es el segundo número es mayor en caso contrario repetir el proceso desde restarle

Terminar

Obtener el factorial de un número:

Precondiciones que el número sea natural y igual o mayor que 1

Si es igual a 1 el factorial es 1

N va ser igual a 1 luego n va a ser igual a  $n+1$  si n es igual al numero del que queremos sacar el factorial se multiplican todos los valores que ha tenido n y el resultado será el factorial deseado en caso contrario repetir proceso desde  $n+1$

Terminar

Los algoritmos fueron modificados varias veces para no dar errores

Prueba de algoritmos;

Proceso:

Determinar si un número es positivo o negativo

$$54 + \sqrt{54^2} = 54 \quad 54 + 54 = 108 \neq 0 \quad \text{positivo}$$
$$-9 + \sqrt{(-9)^2} = -9 \quad -9 + 9 = 0 \quad \text{negativo}$$
$$-14 + \sqrt{(-14)^2} = -14 \quad -14 + 14 = 0 \quad \text{negativo}$$
$$8 + \sqrt{8^2} = 8 \quad 8 + 8 = 16 \neq 0 \quad \text{positivo}$$

0 no cumple

Obtener el mayor de dos números diferentes

(4,5)

$$4-1=3 \neq 5 \neq 0 \quad 3-1=2 \neq 5 \neq 0 \quad 2-1=1 \neq 5 \neq 0 \quad 1-1=0 \quad 5 \text{ es mayor}$$

(-9,6)

16 es mayor

(127, 8+9) no cumple

(7 fm) no cumple

Obtener el Factorial

5

$$n=1 \quad n=1+1=2 \quad n=2+1=3 \neq 5$$

$$n=2+1=3 \neq 5 \quad n=3+1=4 \neq 5 \quad n=4+1=5$$

$$1 \times 2 \times 3 \times 4 \times 5 = 120$$

9

$$n=1 \quad n=1+1=2 \neq 9 \quad n=2+1=3 \neq 9 \quad n=3+1=4 \neq 9 \quad n=4+1=5 \neq 9 \quad n=5+1=6 \neq 9 \quad n=6+1=7 \neq 9$$

$$n=7+1=8 \neq 9 \quad n=8+1=9$$

$$1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8 \times 9 = 362,880$$

0 no cumple

~3 no cumple

## Resultados:

Algoritmo	Entrada	Salida
Determinar si un numero es positivo o negativo	54	54 es positivo
Determinar si un numero es positivo o negativo	-9	-9 es negativo
Determinar si un numero es positivo o negativo	-14	-14 es negativo
Determinar si un numero es positivo o negativo	8	8 es positivo
Determinar si un numero es positivo o negativo	0	No cumple las precondiciones
Obtener el mayor de dos números diferentes:	(4,5)	5 es mayor

Obtener el mayor de dos números diferentes:	(-9,16)	16 es mayor
Obtener el mayor de dos números diferentes:	(127,8+4i)	No cumple con las precondiciones
Obtener el mayor de dos números diferentes:	(7+m)	No cumple con las precondiciones
Obtener el factorial de un número	5	120
Obtener el factorial de un número	9	362,880
Obtener el factorial de un número	0	No cumple las precondiciones
Obtener el factorial de un número	-3	No cumple las precondiciones

Nota:  
 $0! = 1$

Algoritmos para;

Cambiar el signo de un número binario:

Tomar un número de la memoria y pasarlo al registro Z ver el número, identificar de derecha a izquierda si se trata de 1 o 0 por lo mientras ir anotando los dígitos cuando se encuentra un 1 se anota pero el siguiente dígito que se identifique se cambiara a 0 si es 1 y a 1 si es 0 hasta llegar al ultimo número luego pasar los dígitos que anotamos al registro X terminar

Hacer una suma larga binaria:

Precondiciones:  $1+1=0$     $1+0=1$     $0+0=0$

Tomar un número y pasarlo al registro A, tomar otro y pasarlo al registro B, alinear los dígitos del registro A y B empezando con los primeros dígitos de la derecha efectuar una suma vertical cada vez que encontramos la suma  $1+1$  se añadirá un 1 arriba del dígito de la izquierda una vez terminado pasar a los siguientes dígitos de la izquierda y repetir el proceso desde efectuar una suma vertical hasta que ya no haya dígitos a la izquierda poner el numero resultante en el registro C terminar

Aquí se uso un lenguaje mas parecido a la de las maquinas se debe de saber que acciones pueden hacer y que otras no para diseñar un algoritmo paso por paso

# **Conclusión**

Es muy importante saber que es lo que queremos obtener esto se encontraría en la definición de necesidades en el ciclo de vida de software, la importancia se debe porque desde ahí podemos definir nuestras precondiciones para un algoritmo eficaz sin embargo al hacer las pruebas se pueden encontrar con muchos errores que nos llevaría a plantear las precondiciones de nuevo o a modificar el diseño de nuestro algoritmo, es muy útil utilizar el ciclo de vida del software para guiarnos y no dejar de recrear nuestro algoritmo hasta que sea el optimo, también debemos de ser conscientes de lo que se puede o no se puede hacer, por último pero sin menos importancia el algoritmo debe de ser muy minucioso con las instrucciones mas si el algoritmo va a ser para una computadora, se debe de especificar todo para obtener el resultado deseado.