



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Rodríguez Espino Claudia

*Asignatura:* Fundamentos de Programación

*Grupo:* 3

*No de Práctica(s):* Práctica 3: "Solución de problemas y algoritmos"


*Integrante(s):* Borja Portela José Fabio

*Semestre:* 2018-1

*Fecha de entrega:* 9 de marzo del 2018

*Observaciones:*

# CALIFICACIÓN: \_\_\_\_\_

	<b>Manual de prácticas del Laboratorio de Fundamentos de programación</b>	Código:	MADO-17
		Versión:	01
		Página	47/207
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

## Guía práctica de estudio 03: Solución de problemas y Algoritmos.

### Introducción:

En el presente reporte se entregarán evidencias de la realización de la práctica número del calendario semestral correspondiente a la solución de problemas y algoritmos, adjuntando de manera pertinente capturas de pantalla sobre las búsquedas y el desarrollo de las actividades, por otro lado, cada una de estas imágenes será acompañada sobre la información y los pasos sugeridos por el manual de la coordinación, por último, es importante mencionar que todos los eventos aquí demostrados están seguidos de acuerdo a los objetivos, teniéndolos siempre en mente sobre el punto al cual queremos llegar.

### 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

### Desarrollo de las actividades:

1.- Se explicó el ciclo del software de acuerdo a las normas iso.

#### Ciclo de vida del software

La ISO (International Organization for Standardization) en su norma 12207 define al ciclo de vida de un software como:

Un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso.



Figura 1: Ciclo de vida del software.

2.- Se manejó el concepto de algoritmo como método para la solución de problemas y se nos ilustro como un método de entrada de datos, procesamiento de los mismos y por último la salida de datos, se anexa a continuación uno de los ejemplos hechos en laboratorio.

### **Ejemplo 1**

**PROBLEMA:** Determinar si un número dado es positivo o negativo.

**RESTRICCIONES:** El número no puede ser cero.

**DATOS DE ENTRADA:** El conjunto de datos de entrada E está compuesto por el conjunto de los números reales, excepto el cero.

$$E \subset \mathbb{R}^1, \text{ donde} \\ \text{num} \in E \text{ de } (-\infty, \infty) - \{0\}$$

**NOTA:**  $\mathbb{R}^1$  representa al conjunto de números reales de una dimensión.

**DATOS DE SALIDA:** El conjunto de salida S está compuesto por dos valores mutuamente excluyentes.

Un posible conjunto de salida son los valores enteros 0 o 1, donde 0 indica que el valor es positivo y 1 indica el valor es negativo.

$$\text{res} = 0, \text{ si num } (0, \infty), \text{ res} = 1, \text{ si num } (-\infty, 0)$$

Otro posible conjunto de datos de salida son los valores booleanos o lógicos *Verdadero* o *Falso*, donde *Verdadero* indica que el valor es positivo y *Falso* indica que el valor es negativo; o viceversa, *Verdadero* indica que el valor es negativo y *Falso* indica que el valor es positivo.

3.- Posteriormente se manejó el algoritmo como una fase temprana de la etapa de desarrollo de software dándose sus características principales para su identificación, las cuales son listadas a continuación:

- Preciso: Debe indicar el orden de realización de paso y no puede tener ambigüedad.
- Definido: Si se sigue dos veces o más se obtiene el mismo resultado
- Finito: Tiene fin, es decir tiene un número determinado de pasos
- Correcto: Cumplir con el objetivo
- Debe de tener al menos una salida y esta debe de ser perceptible
- Debe de ser sencillo y legible
- Eficiente: Realizarlo en el menor tiempo posible
- Eficaz: que produzca el efecto esperado

4.- Para finalizar se introdujo el concepto de pruebas de escritorio las cuales en pocas palabras es la

ejemplificación de aquellos algoritmos realizados con números o situaciones particulares para así entender cómo es que podría actuar nuestro pseudoprograma, a continuación se muestra el ejemplo 3 visto en clase con algoritmo de solución de problemas y sus pruebas de escritorio.

### Ejemplo 3

**PROBLEMA:** Obtener el factorial de un número dado. El factorial de un número está dado por el producto de ese número por cada uno de los números anteriores hasta llegar a 1. El factorial de 0 (0!) es 1.

**RESTRICCIONES:** El número de entrada debe ser entero y no puede ser negativo.

**DATOS DE ENTRADA:** Número entero.

**DATOS DE SALIDA:** La impresión del factorial del número.

**DOMINIO:** Todos los números naturales positivos.

#### SOLUCIÓN:

1. Solicitar un número entero.
2. Si el número entero es menor a cero regresar al punto 1.
3. Si el número entero es mayor a cero se crea una variable entera *contador* que inicie en 2 y una variable entera *factorial* que inicie en uno.
4. Si la variable contador es menor o igual al número entero de entrada se realiza lo siguiente:
  - 4.1 Se multiplica el valor de la variable *contador* con el valor de la variable *factorial*. El resultado se almacena en la variable *factorial*.
  - 4.2 Se incrementa en uno el valor de la variable *contador*.
  - 4.3 Regresar al punto 4.
5. Si la variable contador no es menor o igual al número entero se muestra el resultado almacenado en la variable *factorial*.

Prueba de escritorio (X es el número entero del que se calculará el factorial):

Iteración	X	factorial	contador	Salida
1	0	1	2	El factorial de 0 es: 1

Iteración	X	factorial	contador	Salida
1	-2	1	2	-
2	-67	1	2	-
3	5	1	2	-
4	5	2	3	-

## Complementos de la práctica

En esta sección se encuentran tres ejemplos más solicitados por la profesora, los cuales fueron desarrollados en casa con el fin de reafirmar conocimientos sobre lo visto en laboratorio, los tres casos y sus números de pruebas de escritorio solicitadas son listados a continuación, para posteriormente comenzar con su desarrollo.

1. Área del círculo (2 pruebas)
2. Formula general desarrollada para casos de raíz positiva y raíz negativa (2 pruebas, una para cada caso)
3. Operaciones preestablecidas para dos casos, cuando se trata de un número mayor a 2 (resolver  $y=x^2-4x+20$ ), otra para menor a 2 (resolver  $y=3x^2+8x+2$ ) y por ultimo una para 2 donde se debe de arrojar un error (3 pruebas, una para cada caso).

### Área del círculo:

PROBLEMA: determinar el área de un círculo

RESTRICCIONES: el radio ingresado no puede ser cero,  $\pi$  es una constante con valor a 3.14

DATOS DE ENTRADA: valor del radio "r"

OPERACIONES: ( $r^2$ ) ( $\pi$ )

DATOS DE SALIDA: área del círculo

ALGORITMO:

1. Inicio
2. Conocer el radio
3. Realizar la operación
4. Mostrar área del círculo en pantalla
5. Fin

PRUEBA DE ESCRITORIO		
Iteración	X	Salida
1	5	78.5
2	7	153.86

### Formula general:

PROBLEMA: determinar el valor de  $x_1$  y  $x_2$ , mediante la resolución de la formula general

RESTRICCIONES: "a" no puede ser cero, si el resultado de la operación dentro de la raíz, es negativo, se factorizara una  $\sqrt{-1}$  dando igual a  $-i$

DATOS DE ENTRADA: valores de las variables a, b, c.

OPERACIONES:

$$S = b^2 - 4ac \quad (-b + S)/2a \quad (-b - S)/2a$$

DATOS DE SALIDA:  $x_1$  y  $x_2$

ALGORITMO:

1. Inicio
2. Declarar variables a, b, c (donde a diferente de cero  $\neq$ )
3. Declarar  $S=b^2-4ac$ , si S mayor igual que cero, ir a paso 4, si S es menor a cero, ir a paso 10.
4. Sacar la raíz cuadrada de S
5. Sumar el resultado de la raíz cuadrada de S a menos b y divide entre dos por a  $(-b+(\text{root}(S)))/(2*a)$
6. Obtén resultado de  $x_1$
7. Resta el resultado de la raíz cuadrada de S a menos b y divide entre dos por a  $(-b-(\text{root}(S)))/(2*a)$



8. Obtén el resultado de  $x_2$
9. Imprime el resultado de  $x_1$  y  $x_2$ , ve a paso 17
10. Si S es menor de cero, se declara valor absoluto de S y se le factoriza una raíz cuadrada de menos uno, la cual es igual a "i"
11. Saca raíz a S y luego multiplícalo por i
12. Suma la raíz de S a menos b, divídelos entre dos por a y posteriormente multiplícalos por i  $[(-b+(\text{root}(S)))/(2*a)]*i$
13. Obtén el resultado de  $x_1$
14. Resta la raíz de S a menos b, divídelos entre dos por a y posteriormente multiplícalos por i  $[(-b-(\text{root}(S)))/(2*a)]*i$
15. Obtén el resultado de  $x_2$
16. Imprime el resultado de  $x_1$  y  $x_2$
17. Fin

PRUEBA DE ESCRITORIO		
Iteración	- "a, b, c"	Salida
1	1, 1, 1	$X_1 = 0.366i$ $X_2 = -1.366i$
2	2, 2, .2	$X_1 = 0.366$ $X_2 = -1.366$

#### Ecuaciones:

PROBLEMA: encontrar el valor para y cuando x es mayor a 2 o menor a 2

RESTRICCIONES: x no puede ser 2

DATOS DE ENTRADA: el valor de la variable 2

OPERACIONES:  $y = x^2 - 4x + 20$     $y = 3x^2 + 8x + 2$

DATOS DE SALIDA: el valor de y para la ecuacion correspondiente

ALGORITMO:

1. Inicio
2. Declarar x con un valor diferente de 2
3. Si  $x=2$ , ir a 4, si no ir a 5
4. Imprime "tu numero debe de ser diferente de 2", ir a paso 2
5. Si x es mayor de 2 ir a 6, si x es menor que 2 ir a paso 9
6. Realiza  $x^2 - 4x + 20$
7. Declara  $y = x^2 - 4x + 20$
8. Imprime valor de y, ve a paso 12
9. Realiza  $3x^2 + 8x + 2$
10. Declara  $y = 3x^2 + 8x + 2$
11. Imprime resultado de y
12. Fin

PRUEBA DE ESCRITORIO		
Iteración	x	Salida
1	2	Tu número debe de ser diferente de 2
2	1	13
3	3	41

## Conclusiones

Después de haber realizado las actividades, se ha aprendido más sobre el manejo de los algoritmos y su importancia a la hora de programar.