



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Karina

*Asignatura:* Fundamentos de Programación

*Grupo:* 20

*No. de práctica(s):* No. 3 Solución de problemas y algoritmos.

*Integrante(s):* López Olmedo Ernesto Yael

*No. de lista o brigada:* 27

*Semestre:* 2024-1

*Fecha de entrega:* 12 de Septiembre 2023

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

## **Introducción**

Para la tercer práctica de laboratorio el alumno ahora conocerá la metodología de solución de problemas y la creación de algoritmos, tras el conocimiento de la formulación de resoluciones de manera cualitativa y de la forma cuantitativa, ahora deberá de integrar estas metodologías a un procedimiento más preciso y más orientado a lo que serán las bases de la formación y las componentes de un programa computacional.

## **Objetivo**

El alumno elabore algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida de un software.

## Desarrollo

Para el desarrollo debemos conocer lo siguientes puntos:

- Ciclo de vida de un software.
- Solución de problemas.
- Algoritmos.

A - Ciclo de vida de un software:

“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 1.1 -Ciclo de vida de un software.

B - Solución de problemas:

El análisis es el proceso para averiguar qué es lo que requiere el usuario del sistema de software (análisis de requisitos). Esta etapa permite definir las necesidades de forma clara y concisa (especificación de requisitos). Por lo tanto, la etapa del análisis consiste en conocer qué es lo que está solicitando el usuario. Para ello es importante identificar dos grandes conjuntos dentro del sistema: el conjunto de entrada y el conjunto de salida. En la Figura 2 se muestra de manera esquemática los componentes del sistema

Los datos de entrada están compuestos por todos aquellos datos que pueden alimentar al sistema; los conjuntos de salida están compuestos por todos los datos que el sistema regresará como resultado del proceso. Estos datos se obtienen a partir de los datos de entrada.

La unión del conjunto de entrada y el conjunto de salida forman lo que se conoce como el dominio del problema, es decir, los valores que el problema puede manejar.

La etapa de análisis es crucial para la creación de un software de calidad, ya que si no se entiende qué es lo que se desea realizar, no se puede generar una solución. Sin embargo,

es común caer en ambigüedades debido al mal entendimiento de los requerimientos iniciales

## C - Algoritmos:

Un algoritmo se define como un conjunto de reglas, expresadas en un lenguaje específico, para realizar alguna tarea en general, es decir, un conjunto de pasos, procedimientos o acciones que permiten alcanzar un resultado o resolver un problema.

Estas reglas o pasos pueden ser aplicados un número ilimitado de veces sobre una situación particular. Un algoritmo es la parte más importante y durable de las ciencias de la computación debido a que éste puede ser creado de manera independiente tanto del lenguaje como de las características físicas del equipo que lo va a ejecutar.

Las principales características con las que debe cumplir un algoritmo son:

- 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 tener al menos una salida y ésta debe de ser perceptible
- Debe ser sencillo y legible
- Eficiente: Realizarlo en el menor tiempo posible
- Eficaz: Que produzca el efecto esperado

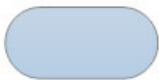

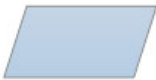


Símbolo	Nombre	Función
	Inicio / Final	Representa el inicio y el final de un proceso
	Línea de Flujo	Indica el orden de la ejecución de las operaciones. La flecha indica la siguiente instrucción.
	Entrada / Salida	Representa la lectura de datos en la entrada y la impresión de datos en la salida
	Proceso	Representa cualquier tipo de operación
	Decisión	Nos permite analizar una situación, con base en los valores verdadero y falso

Figura 1.2 - Representación de algoritmos.

### Ejemplo 1:

Problema: Determinar si un número dado es positivo o negativo.

Restricciones: El número no puede ser cero.

Datos de entrada: Número real.

Datos de salida: La indicación de si el número es positivo o negativo.

Dominio: Todos los números reales.

#### *Solución:*

- 1 - Crear las variables num1 con valor 0;
- 2 - Imprimir en pantalla: "Ingrese sus valores" y sustituir en num1;
- 3 - Si num1 es igual a 0, se reinicia el proceso;
- 4 - Hacer el proceso: en num1;
  - 4.1 - Si  $\text{num1} < 0$ : Es "Negativo", se almacena el valor;
  - 4.2 - Si  $\text{num1} > 0$ : Es "Positivo" se almacena el valor;
- 5 - Presentar en pantalla: "El número es: " + resultado paso 4;

#### *Prueba de escritorio:*

Iteración	X	Salida
1	5	El número es: Positivo
2	-13	El número es: Negativo
3	6689	El número es: Positivo

## Ejemplo 2:

Problema: Obtener el mayor de dos números dados.

Restricciones: Los números de entrada deben ser diferentes.

Datos de entrada: Dos números reales.

Datos de salida: La escritura del número más grande. DOMINIO: Todos los números reales.

### *Solución:*

- 1 - Solicitar entrada real como var1, con valor 0 e imprimir "Ingresa tu primer valor";
- 2 - Solicitar entrada real como var2, con valor 0 e imprimir "Ingresa tu segundo valor";
- 3 - Comprobar que  $val1 \neq val2$ ;
- 4 - Si  $val1 > val2$ :
  - 4.1 - Presentar val1 y " es el número mayor";
- 5 - Si  $val1 < val2$ :
  - 5.1 - Presentar val2 y "es el número mayor "

### *Prueba de escritorio:*

Iteración	val1	val2	Salida
1	5	5	-
2	3	6	6 es el número mayor
3	4	-5	4 es el número mayor

### Ejemplo 3:

Problema: Obtener el factorial de un número dado.

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

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

Datos de entrada: Número entero.

Datos de salida: El factorial del número.

Dominio: Todos los números naturales y el cero.

#### *Solución:*

- 1 - Solicitar un número entero y guardarlo en una variable llamada numInt.
- 2 - Si numInt es menor a cero regrese al punto 1.
- 3 - Si numInt es mayor o igual a cero se crea una variable entera contador que inicie en 2 y una variable entera factorial que inicie en 1.
- 4 - Si la numInt es menor o igual al número entero de entrada se realiza lo siguiente:
  - 4.1 Se multiplica numInt con el valor de la variable factorial. El resultado se almacena en la variable factorial.
  - 4.2 Se incrementa en uno numInt.
- 5 - Si numInt es menor o igual al número entero de entrada se muestra el resultado almacenado en la variable factorial  
"El resultado es: "

#### *Pruebas de escritorio:*

Iteración	numInt	factorial	contador	Resultado
1	5	1	2	—
2	5	120	6	El resultado es: 120
3	0	1	2	El resultado es: 1

## Análisis de problemas

### Problema 1:

PROBLEMA: Seguir el algoritmo para obtener una figura

ENTRADA: Hoja tamaño carta en limpio, regla y lápiz.

SALIDA: Figura correcta

#### *Retroalimentación:*

No contiene precisión en la forma de dar las instrucciones, son muy vagas, ya que una línea puede variar, siendo que solo se menciona mediante fracciones, que realmente no contempla la forma en que el lector la interpretará. Otra parte los datos no están descritos, ya que una figura puede resultar de diferentes formas, sería más sencillo si es que se creará un menú que contemple:

- Formato de la hoja, para hacer las adecuaciones.
- Preestablecer ciertas figuras como triángulo, cuadrado, rombo, entre otras más.
- Medidas por ejemplo: con grados, dimensiones en cm.
- Especificar los instrumentos a usar.

### Problema 2:

PROBLEMA: Seguir el algoritmo para obtener una figura

ENTRADA: Hoja tamaño carta en limpio, regla y lápiz.

SALIDA: Figura correcta.

#### *Retroalimentación:*

Este presenta una mejor precisión del lenguaje para la creación de una figura en este caso en una estrella, que a través de un compás se trazan seis círculos con mismo radio, que van a estar juntos, que al final de ser trazados son unidos con líneas para formar la estrella, mediante triángulos. Con esa paráfrasis se puede resolver de forma adecuada la creación de una figura, no tiene medidas, pero eso se puede entender que es a condición al que lo desarrolle, eso es lo que lo vuelve eficiente al programa, porque las cosas o ciertas variables no especificadas, son a consideración del desarrollador, no por una falla del programa.



## Tarea

Realiza los dos ejercicios de las figuras en las hojas de tarea, compara y explica cuál es la diferencia entre los dos algoritmos, realiza una crítica constructiva, esto va al final de la práctica antes de la conclusión y bibliografía.

### 1-Temperaturas.

Problema: Realiza un algoritmo que convierta de grados C° a F°.

Restricciones: Tiene que ser numérico real,

Datos de entrada: Operación a hacer, valor de temperatura

Datos de salida: temperatura(trans1)

*Solución:*

1- Concatenar las operaciones a través de las variables String con valor de operación: Trans1 = (C° a F°) & temp(valor de entrada) :

$$\text{Trans1} = (\text{temp} * 1.8) + 32;$$

2- Concatenar las variables C° con valor 0 y F° con valor 0;

3 - Pedir el valor de la temperatura (temp) a transformar.

“Ingrese su temperatura en Celsius”

4 - Hacer la operación correspondiente y almacenarla en Trans1

5 - Presentar en pantalla:

“Su temperatura es: ”+ Trans1 + “grados fahrenheit”

*Pruebas de escritorio:*

Iteraciones	Temperatura C°	Resultado
1	-56	Su temperatura es: -68.8 grados fahrenheit
2	1250	Su temperatura es: 2282 grados fahrenheit
3	22	Su temperatura es: 71.6 grados fahrenheit

## 2- Ecuaciones.

Problema: Realiza un algoritmo para resolver una ecuación de segundo grado.

Restricciones: Tiene que ser numérico, se debe preestablecer la fórmula, debe considerar el operador raíz cuadrada.

Datos de entrada: Números reales, enteros

Datos de salida: Dos números reales (x1 y x2)

### *Solución:*

1 - Se preestablece la fórmula :

$$x1 = -b + \sqrt{b^2 - 4 * a * c} / (2 * a);$$

$$x2 = -b - \sqrt{b^2 - 4 * a * c} / (2 * a);$$

2 - Se inicia el programa con impresión de pantalla:

“Ingrese su valor a: ”

2.1 - Si  $a == 0$ , regresa a paso 2;

3 - Se presenta en la pantalla:

“Ingrese su valor b: ”

3.1 - Si  $b == 0$ , regresa a paso 2;

4 - Se presenta en la pantalla:

“Ingrese su valor c: ”

4.1 - Si  $c == 0$ , regresa a paso 2;

5 - Insertar valores a la fórmula ya concatenada.

6 - En caso de tener valor negativo dentro sqrt, mostrar:

“Error, indefinida”

6.1 - Regresa a paso 2;

7 - Presentar en pantalla:

“Los valores de x son: ” + x1 + “,” + x2;

Pruebas de escritorio:

Iteración	a	b	c	Resultado
1	6	-19	7	Los valores de x son: 2.74, 0.42
2	5	-15	-50	Los valores de x son: 5, -2
3	1	7	12	Los valores de x son: 9, -2

## **Conclusión**

Durante esta práctica el alumno comprendió como precisar sus palabras para la resolución de problemas, tanto matemáticos, como lógicos. Creó metodologías que le permitieran indicar el orden de realización de paso evitando generar ambigüedades, hacerlos finitos mediante el menor número de pasos posibles, siendo entendibles y legibles, para cumplir con tiempos cortos, con el objetivo y el resultado sea el esperado obtener. Todo esto para que el alumno comprenda los temas futuros, que empiece a fomentar la formulación lógica, que se es requerida en la programación, y así pueda crear de manera escrita/gráfica el desarrollo de programas de forma exitosa.

## Referencias

- Templos, A.(2022).Manual de prácticas del laboratorio de Fundamentos de programación. México: UNAM.