



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de Docencia

Laboratorios de Computación Salas A y B

Profesor: **Ing. Karina García Morales**

Asignatura: **Fundamentos de Programación**

Grupo: **22**

No. de práctica(s): **Práctica 3. Solución de Problemas y Algoritmos.**

Integrante(s): **Edilberto Vicente Martínez**

No. de lista o brigada: **49**

Semestre: **2026-1**

Fecha de entrega: **Martes 23 de septiembre de 2025**

Observaciones:

CALIFICACIÓN: _____

Solución de Problemas y Algoritmos

Objetivo:

- El alumno elaborará 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 la práctica:

Un problema informático se puede definir como el conjunto de instancias al cual corresponde un conjunto de soluciones, junto con una relación que asocia para cada instancia del problema un subconjunto de soluciones (posiblemente vacío).

Para poder solucionar un problema nos apoyamos en la Ingeniería de Software que de acuerdo con la IEEE se define como “la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software”. Por lo que el uso y establecimiento de principios de ingeniería sólidos, son básicos para obtener un software que sea económicamente fiable y funcione eficientemente.

La Ingeniería de Software provee métodos que indican cómo generar software. Estos métodos abarcan una amplia gama de tareas:

- Planeación y estimación del proyecto.
- Análisis de requerimientos del sistema y software.
- Diseño de la estructura de datos, la arquitectura del programa y el procedimiento algorítmico.
- Codificación.
- Pruebas y mantenimiento (validación y verificación).

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”

La Figura 1 muestra las actividades, también nombradas etapas, que se realizan en el ciclo de vida del software.



Figura 1. Ciclo de vida del software (Facultad de Ingeniería, 2025).

Solución de problemas:

Dentro del ciclo de vida del software, en el análisis se busca comprender la necesidad, es decir, entender el problema.

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.

El conjunto de entrada está compuesto por todos aquellos datos que pueden alimentar al sistema.

El conjunto de salida está compuesto 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.



Figura 2. Sistema (Facultad de Ingeniería, 2025).

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.

Algoritmos:

Una vez realizado el análisis, es decir, ya que se entendió qué es lo que está solicitando el usuario y ya identificado el conjunto de entrada y el conjunto de salida, se puede proceder al diseño de la solución, esto es, a la generación del algoritmo.

Dentro del ciclo de vida del software, la creación de un algoritmo se encuentra en la etapa de diseño. Ver figura 1.

Durante el diseño se busca proponer una o varias alternativas viables para dar solución al problema y con base en esto tomar la mejor decisión para iniciar la construcción.

Un problema matemático es computable si éste puede ser resuelto, en principio, por un dispositivo computacional.

La teoría de la computabilidad es la parte de la computación que estudia los problemas de decisión que pueden ser resueltos con un algoritmo.

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.

Por tanto, un buen algoritmo debe ser correcto (cumplir con el objetivo) y eficiente (realizarlo en el menor tiempo posible), además de ser entendible para cualquier persona.

Las actividades a realizar en la elaboración de un algoritmo para obtener una solución a un problema de forma correcta y eficiente se muestran en la Figura 6.

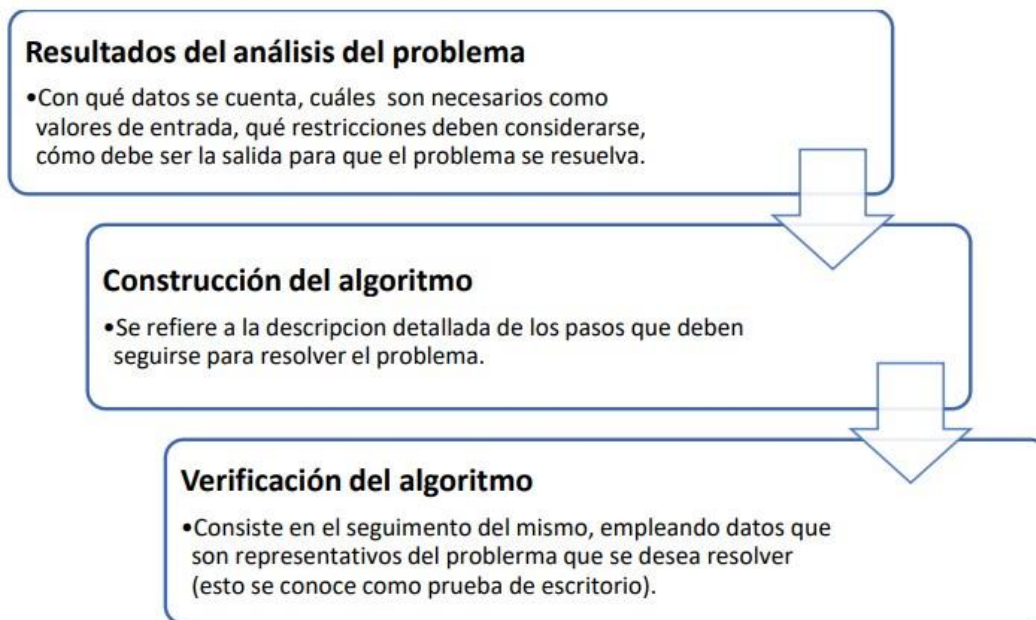


Figura 3. Elaboración de un algoritmo (Facultad de Ingeniería, 2025).

Un algoritmo consta de 3 módulos básicos (Figura 4):

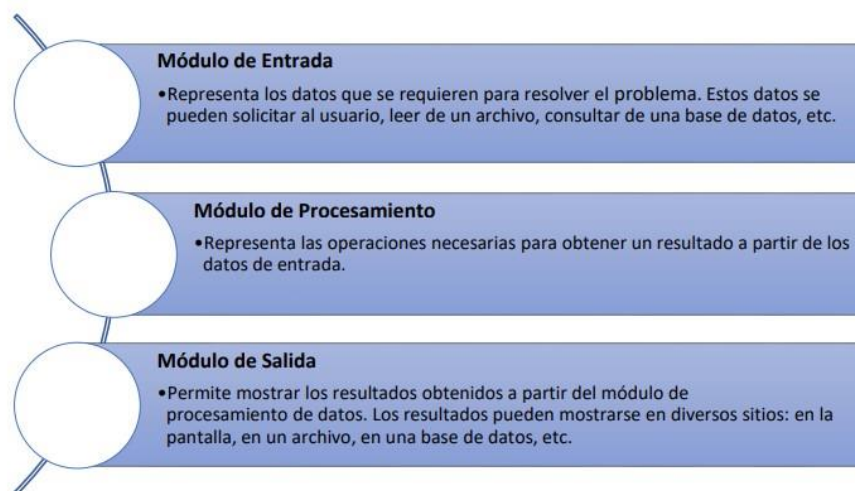


Figura 4. Módulos básicos del algoritmo (Facultad de Ingeniería, 2025).

Variables:

Un algoritmo requiere del uso de variables porque guardan el valor (numérico o no numérico) de los datos de entrada; también suelen ser utilizadas para almacenar datos generados en el proceso y datos de salida. Es a través del valor de dichas variables que el algoritmo puede fluir en la secuencia de pasos a seguir.

Prueba de escritorio:

El diseño de la solución de un problema implica la creación del algoritmo y la validación de este. La validación se suele realizar mediante una prueba de escritorio.

Una prueba de escritorio es una matriz formada por los valores que van adquiriendo cada una de las variables del algoritmo en cada iteración. Una iteración es el número de veces que se

ejecuta una parte del algoritmo y permite ver los valores que van adquiriendo las variables en cada repetición.

Ejercicio de tarea:

1. Realiza los dos algoritmos de la práctica e indica que sugieres u observas en cada uno de los algoritmos.

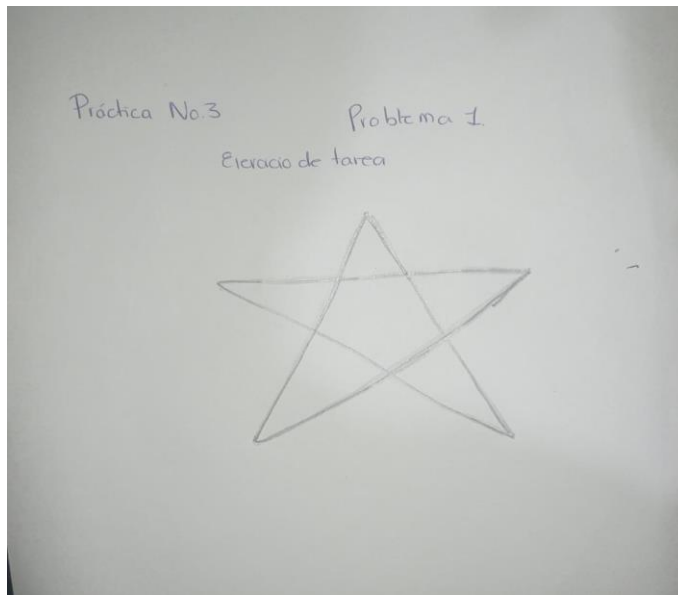
PROBLEMA: Seguir el algoritmo para obtener una figura.

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

ALGORITMO:

1. Dibuja una V invertida. Empieza desde el lado izquierdo, sube, y baja hacia el lado derecho, no levantes el lápiz.
2. Ahora dibuja una línea en ángulo ascendente hacia la izquierda. Debe cruzar la primera línea más o menos a $1/3$ de la altura. Todavía no levantes el lápiz del papel.
3. Ahora, dibuja una línea horizontal hacia la derecha. Debe cruzar la V invertida más o menos a $2/3$ de la altura total. Sigue sin levantar el lápiz.
4. Dibuja una línea en un ángulo descendente hasta el punto de inicio. Las líneas deben unirse.
5. Ahora ya puedes levantar el lápiz del papel. Has terminado la estrella de 5 puntas.

EVIDENCIA:



El realizar este algoritmo me pareció sencillo, ya que se entendían muy bien los pasos, solamente que considero que este algoritmo no es el más preciso para poder realizar una

estrella de 5 puntas por el hecho de que al no poder despegar el lápiz es muy probable que nuestras líneas no sean lo más rectas posibles caso contrario al segundo algoritmo.

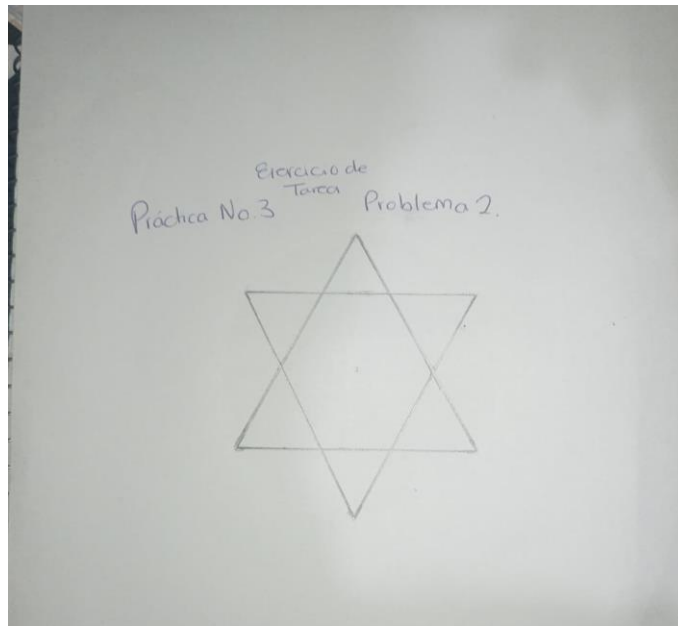
PROBLEMA: Seguir el algoritmo para obtener una figura.

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

ALGORITMO

1. Empieza dibujando un círculo con un compás. Coloca un lápiz en el compás. Coloca la punta del compás en el centro de una hoja de papel.
2. Ahora gira el compás, mientras mantienes la punta apoyada en el papel. El lápiz dibujará un círculo perfecto alrededor de la punta del compás.
3. Marca un punto en la parte superior del círculo con el lápiz. Ahora, coloca la punta del compás en la marca. No cambies el radio del compás con que hiciste el círculo.
4. Gira el compás para hacer una marca en el propio círculo hacia la izquierda. Haz una marca también en el lado derecho.
5. Ahora, coloca la punta del compás en uno de los puntos. Recuerda no cambiar el radio del compás. Haz otra marca en el círculo.
6. Continúa moviendo la punta del compás a las otras marcas, y continúa hasta que tengas 6 marcas a la misma distancia unas de otras. Ahora, ya puedes dejar tu compás a un lado.
7. Usa una regla para crear un triángulo que empiece en la marca superior del círculo. Coloca el lápiz en la marca superior. Ahora dibuja una línea hasta la segunda marca por la izquierda. Dibuja otra línea, ahora hacia la derecha, saltándote la marca de la parte más baja. Complementa el triángulo con una línea hacia la marca superior. Así completarás el triángulo.
8. Crea un segundo triángulo empezando en la marca en la base del círculo. Coloca el lápiz en la marca inferior. Ahora conéctala con la segunda marca hacia la izquierda. Dibuja una línea recta hacia la derecha, saltándote el punto superior. Completa el segundo triángulo dibujando una línea hasta la marca en la parte inferior.
9. Borra el círculo. Has terminado de dibujar tu estrella de 6 puntos.

EVIDENCIA:



Al realizar este segundo algoritmo se pudo notar una mayor precisión y mejor desarrollo de la figura, aunque sea un poco más difícil de seguir las instrucciones creo que es un muy buen algoritmo para el desarrollo del problema y considero que a veces es muy bueno apoyarse de herramientas para la elaboración de nuestra figura, como es el caso de la regla y el compás.

2. Repetir ejercicio 3 de desarrollo para 5 alumnos: Solicitar 3 calificaciones, obtener el promedio, si es mayor igual a 6 mostrar "Aprobado", en caso contrario mostrar "Reprobado".

PROBLEMA: Obtener el promedio de 3 calificaciones parciales de una materia, para verificar que 5 alumnos diferentes la hayan aprobado.

RESTRICCIONES: Solo se pueden introducir números del 0 al 10.

DATOS DE ENTRADA: 3 número entre 0 y 10.

DATOS DE SALIDA: "Aprobado" o "Reprobado".

DOMINIO: Todos los números naturales y el cero.

SOLUCIÓN/ALGORITMO:

1. Generar una variable para almacenar la suma de los números.
2. Generar una variable para almacenar el promedio.
3. Generar una variable para almacenar los números ingresados.
4. Generar una variable para contar la cantidad de números ingresados.
5. Solicitar primer número y almacenarlo en una variable (calif1).
6. Solicitar segundo número y almacenarlo en una variable (calif2).

7. Solicitar tercer número y almacenarlo en una variable (calif3).

7.1. Si el numero ingresado es menor a cero regresar al paso 5.

8. Sumar los tres números ($\text{calif1} + \text{calif2} + \text{calif3} = \text{PROM}$).

9. Dividir PROM entre 3 ($\text{PROM}/3 = \text{califFINAL}$).

10. Si $\text{califFINAL} \geq 6$, se realiza lo siguiente:

10.1 Si $\text{califFINAL} \geq 6$, se muestra “APROBADO”.

10.2 Si $\text{califFINAL} < 6$, se muestra “REPROBADO”.

PRUEBA(S) DE ESCRITORIO:

Iteración/Alumno	calif1	calif2	calif3	PROM	califFINAL	Salida
1	7.5	7.9	6.4	21.8	7.26	APROBADO
2	8.5	9.3	7.6	25.4	8.46	APROBADO
3	5	6	6.3	17.3	5.76	REPROBADO
4	10	9.8	10	29.8	9.93	APROBADO
5	8.8	7.6	9.7	26.1	8.7	APROBADO

Conclusiones:

Al haber terminado esta práctica considero que he aprendido un poco más sobre la importancia de los algoritmos en nuestra vida, ya que gracias a ellos se ha podido realizar grandes avances en la tecnología, es por ello por lo que considero que el tener, aunque sea una mínima idea de cómo funcionan los algoritmos en un software es muy importante porque nos sirve a resolver problemas y ser más eficientes y seguros con nuestros resultados, lo cual es muy importante para cualquier ingeniero.

Además, el haber realizado estos ejercicios demuestran que sin darnos cuenta muchas veces necesitamos de algoritmos en la vida cotidiana para poder realizar tareas o actividades aprovechando menos recursos y menos tiempo.

Fuentes de consulta:

- Facultad de Ingeniería. (2025). Manual de prácticas del laboratorio de Fundamentos de Programación. Laboratorio de computación. Salas A y B. Universidad Nacional Autónoma de México. pp. 36-51. Recuperado el 23 de septiembre de 2025 de <http://lcp02.fi-b.unam.mx/>