



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Marco Antonio Martínez Quintana

Asignatura: Fundamentos de Programación

Grupo: 3

No de Práctica(s): 3

Integrante(s): Ramírez Pérez Daniela Itzel

*No. de Equipo de cómputo
empleado:* No aplica

No. de Lista o Brigada: 36

Semestre: 1er

Fecha de entrega: Viernes 23 de octubre

Observaciones:

CALIFICACIÓN: _____

Solución de problemas y Algoritmos

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.

Introducción

A un problema de computadora siempre tendrá como equivalencia un conjunto y subconjunto de soluciones

Para solucionar los problemas debemos ver a la Ingeniería de Software que es la aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software.

Ciclo de vida del 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: Ciclo de vida del software.

Solución de problemas

En el análisis de problema se busca comprenderlo para averiguar qué es lo que requiere el usuario del sistema de software

Para ello es importante identificar dos grandes conjuntos dentro del sistema: el conjunto de entrada y el conjunto de salida.

- Conjunto de entrada: Datos que alimentan al sistema.
- Conjunto de salida: Datos que el sistema regresará como resultado del proceso, obtenidos a partir de los datos de entrada.

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



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.

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

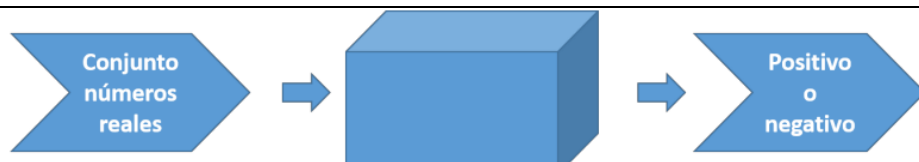
NOTA: \mathbb{R} representa al conjunto de números reales de una dimensión.

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.

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

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.



Ejemplo 2

PROBLEMA: Obtener el mayor de dos números diferentes dados.

RESTRICCIONES: Los números de entrada deben ser diferentes.

DATOS DE ENTRADA: El conjunto de entrada E está dividido en dos subconjuntos E y E' .

El primer número (num1) puede adquirir cualquier valor del conjunto de los números reales ($E = (-\infty, \infty)$), sin embargo, el conjunto de entrada del segundo número (num2) es un subconjunto de E , es decir, E' está compuesto por el conjunto de los números reales excepto num1 ($E' = (-\infty, \infty) - \{\text{num1}\}$).

$E, E' \subset \mathbb{R}$, donde
 $\text{num1} \in E$ de $(-\infty, \infty)$,
 $\text{num2} \in E'$
de $(-\infty, \infty) - \{\text{num1}\}$

DATOS DE SALIDA: El conjunto de datos de salida S que puede tomar el resultado r está compuesto por el conjunto de los números reales.

$S \subset \mathbb{R}$, donde $r \in S$ de $(-\infty, \infty)$



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:

$$n! = n * (n-1)!$$

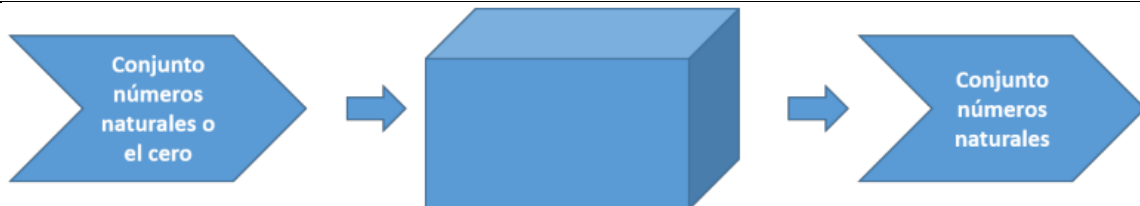
RESTRICCIONES: El número de entrada debe ser entero positivo o cero. No puede ser negativo.

DATOS DE ENTRADA: El conjunto de entrada E está dado por el conjunto de los números naturales o por el cero.

$$E \subset \mathbb{N}_1, \text{ donde} \\ \text{num} \in E \text{ de } [1, \infty) \cup \{0\}$$

DATOS DE SALIDA: El conjunto de salida S está conformado por el conjunto de los números naturales.

$$S \subset \mathbb{N}_1; \text{ donde} \\ \text{res} \in S \text{ de } [1, \infty)$$



Algoritmos

Cuando se realiza el análisis del problema se procede al diseño de la solución, es decir al algoritmo que se encuentra en la etapa del diseño del ciclo de vida del software.

Un algoritmo se define como un conjunto de reglas, 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:

- x Preciso: Debe indicar el orden de pasos y no tiene ambigüedad
- x Definido: Siempre se llega al mismo resultado.
- x Finito: Número determinado de pasos.
- x Correcto: Cumplir con el objetivo.
- x Eficiente: Realizarse en el menor tiempo posible
- x Eficaz: Que produzca el efecto esperado

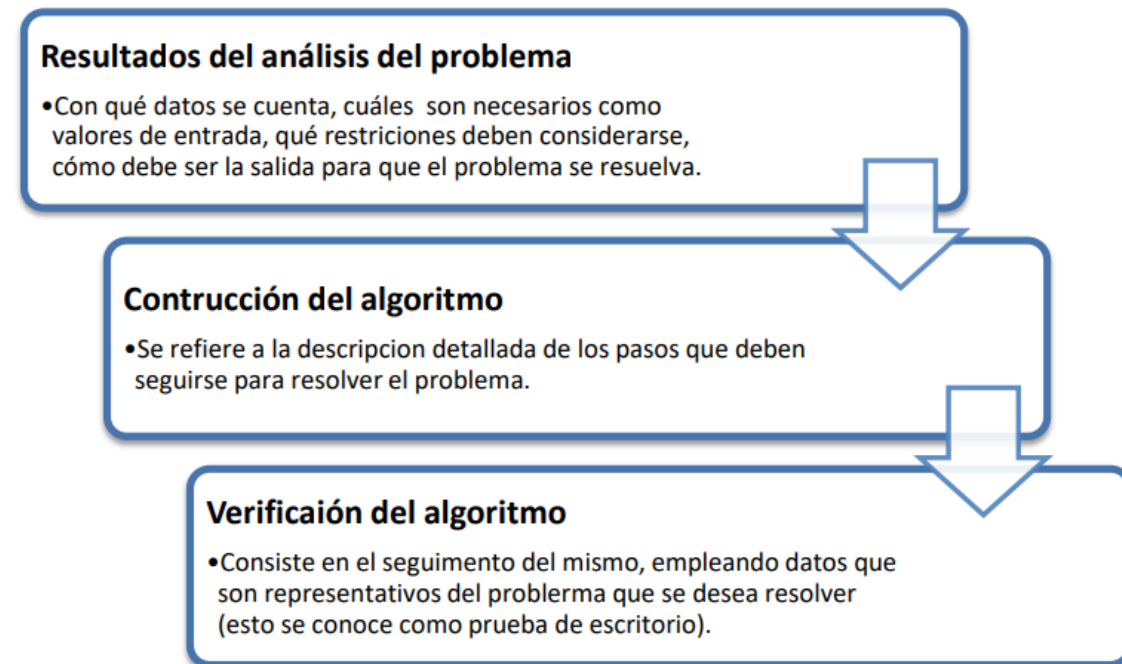
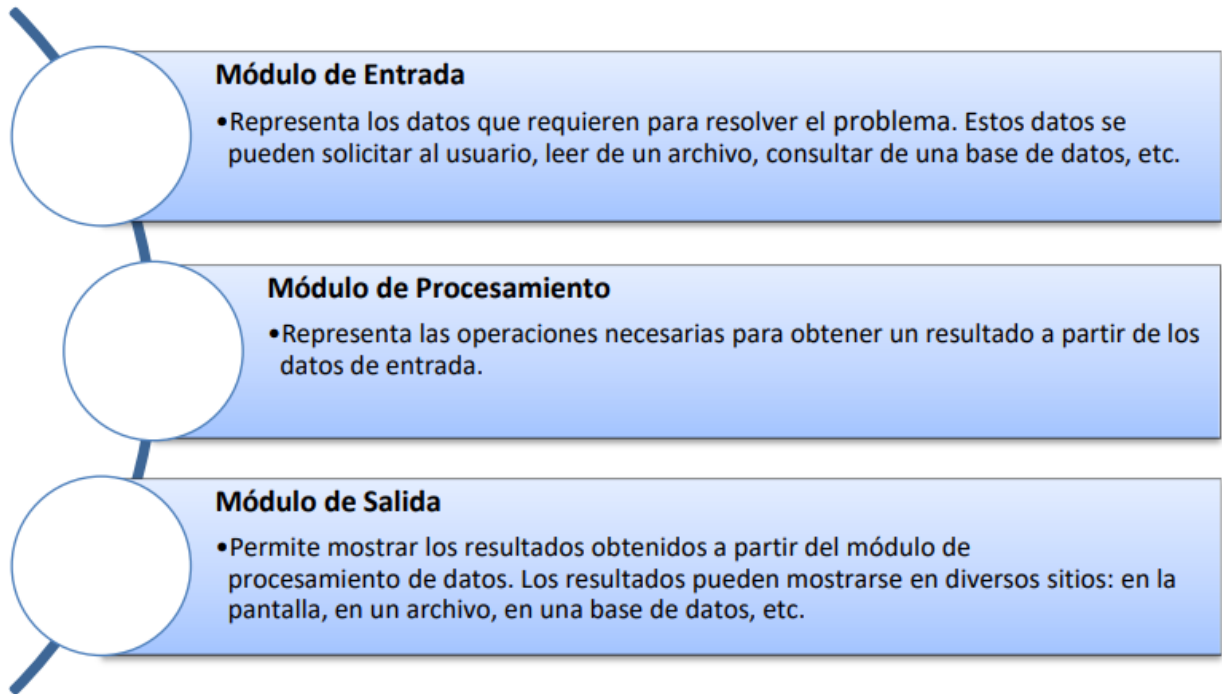


Figura 2: Elaboración de un algoritmo

Un algoritmo consta de 3 módulos básicos:



Ejemplo 1	Prueba de escritorio																											
<p>PROBLEMA: Determinar si un número dado es positivo o negativo.</p> <p>RESTRICCIONES: El número no puede ser cero.</p> <p>DATOS DE ENTRADA: Número real.</p> <p>DATOS DE SALIDA: La validación de si el número es positivo</p> <p>DOMINIO: Todos los números reales.</p> <p>SOLUCIÓN:</p> <ol style="list-style-type: none">Solicitar un número real.Si el número ingresado es cero, se regresa al punto 1.Si el número ingresado es diferente de cero, se validan las siguientes condiciones:<ol style="list-style-type: none">Si el número ingresado es mayor a 0 se puede afirmar que el número es positivo.Si el número ingresado es menor a 0 se puede afirmar que el número es negativo.	<table><tr><th>Iteración</th><th>X</th><th>Salida</th></tr><tr><td>1</td><td>5</td><td>El número 5 es positivo</td></tr></table> <table><tr><th>Iteración</th><th>X</th><th>Salida</th></tr><tr><td>1</td><td>-29</td><td>El número 29 es negativo</td></tr></table> <table><tr><th>Iteración</th><th>X</th><th>Salida</th></tr><tr><td>1</td><td>0</td><td>-</td></tr><tr><td>2</td><td>0</td><td>-</td></tr><tr><td>3</td><td>0</td><td>-</td></tr><tr><td>4</td><td>100</td><td>El número 100 es positivo</td></tr></table>	Iteración	X	Salida	1	5	El número 5 es positivo	Iteración	X	Salida	1	-29	El número 29 es negativo	Iteración	X	Salida	1	0	-	2	0	-	3	0	-	4	100	El número 100 es positivo
Iteración	X	Salida																										
1	5	El número 5 es positivo																										
Iteración	X	Salida																										
1	-29	El número 29 es negativo																										
Iteración	X	Salida																										
1	0	-																										
2	0	-																										
3	0	-																										
4	100	El número 100 es positivo																										

Ejemplo 2	Prueba de escritorio																																				
<p>PROBLEMA: Obtener el mayor de dos números dados.</p> <p>RESTRICCIONES: Los números de entrada deben ser diferentes.</p> <p>DATOS DE ENTRADA: Número real.</p> <p>DATOS DE SALIDA: La impresión del número más grande.</p> <p>DOMINIO: Todos los número reales.</p> <p>SOLUCIÓN:</p> <p>1. Solicitar un primer número real.</p> <p>2. Solicitar un segundo número real.</p> <p>3. Si el segundo número real es igual al primer número real, se regresa al punto 2.</p> <p>4. Si el segundo número real es diferente al primer número real, se validan las siguientes condiciones:</p> <p>4.1 Si se cumple con la condición de que el primer número es mayor al segundo número, entonces se puede afirmar que el primer número es el mayor de los números.</p> <p>4.2 Si se cumple con la condición de que el segundo número es mayor al primer número, entonces se puede afirmar que el segundo número es el mayor de los números.</p> <p>Prueba de escritorio (X es el primer número solicitado, Y es el segundo):</p>	<table><tr><th>Iteración</th><th>X</th><th>Y</th><th>Salida</th></tr><tr><td>1</td><td>5</td><td>6</td><td>El número 6 es el mayor</td></tr></table> <table><tr><th>Iteración</th><th>X</th><th>Y</th><th>Salida</th></tr><tr><td>1</td><td>-99</td><td>-222.2</td><td>El número -99 es el mayor</td></tr></table> <table><tr><th>Iteración</th><th>X</th><th>Y</th><th>Salida</th></tr><tr><td>1</td><td>15</td><td>15</td><td>-</td></tr><tr><td>2</td><td>15</td><td>15</td><td>-</td></tr><tr><td>3</td><td>15</td><td>15</td><td>-</td></tr><tr><td>4</td><td>15</td><td>10</td><td>El número 15 es el mayor</td></tr></table>	Iteración	X	Y	Salida	1	5	6	El número 6 es el mayor	Iteración	X	Y	Salida	1	-99	-222.2	El número -99 es el mayor	Iteración	X	Y	Salida	1	15	15	-	2	15	15	-	3	15	15	-	4	15	10	El número 15 es el mayor
Iteración	X	Y	Salida																																		
1	5	6	El número 6 es el mayor																																		
Iteración	X	Y	Salida																																		
1	-99	-222.2	El número -99 es el mayor																																		
Iteración	X	Y	Salida																																		
1	15	15	-																																		
2	15	15	-																																		
3	15	15	-																																		
4	15	10	El número 15 es el mayor																																		

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

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	-

5	5	6	4	-
6	5	24	5	-
7	5	120	6	El factorial de 5 es: 120

Iteración	X	factorial	contador	Salida
1	7	1	2	-
2	7	2	3	-
3	7	6	4	-
4	7	24	5	-
5	7	120	6	-
6	7	720	7	-
7	7	5040	8	El factorial de 7 es: 5040

Ejercicio 1

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 $\frac{1}{3}$ 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 $\frac{2}{3}$ 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.



Ejercicio 2

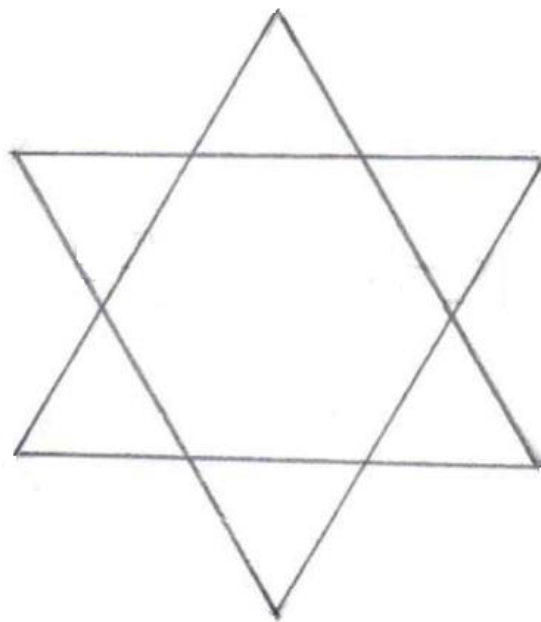
PROBLEMA: Seguir el algoritmo para obtener una figura

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.



Conclusión

Al finalizar esta práctica se tiene una mejor comprensión en cómo los algoritmos nos ayudan en resolver problemas y alcanzar objetivos predeterminados por estos.

Se llega a entender que este tema es de vital importancia al ser una de las características más importantes del ciclo de vida de un software que, sin el mantenimiento del software creado, el software en si se volverá obsoleto cuando se requiera hacer un cambio o actualización.