



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: 20

No. de práctica(s): 03

Integrante(s): Martinez Ordoñez Diego Tonatiah

No. de lista o brigada: 30

Semestre: Primero

Fecha de entrega: 21/ septiembre/ 2022

Observaciones:

CALIFICACIÓN: _____

Práctica 03: 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:

Durante el transcurso de la navegación por internet o dentro del mismo sistema, podemos experimentar los llamados PROBLEMAS INFORMATICOS; el cual se puede definir como un conjunto de instancias y criterios, que te implican un conjunto de soluciones, correspondiendo que a cada instancia, para un subconjunto de solución.

Existen diversos métodos para poder generar un software, los cuales abarcan diversas acciones a realizar, como las siguientes:

- 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”.



- Solución de problemas:

La solución de problemas busca soluciones para poder hallar soluciones que le vayan bien a una situación, dentro de la búsqueda de soluciones encontramos el “análisis”.

El análisis es un proceso para poder saber qué es lo que requiere el usuario del sistema de software. Esta etapa permite definir las necesidades que presenta el usuario, siendo precisas y claras.

Para lo cual debemos identificar los 2 conjuntos que están dentro del sistema:

1. El conjunto de entrada está compuesto por todos aquellos datos que pueden alimentar al sistema.
2. 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. Estos 2 son fundamentales, ya que son los valores que pueden manejar dentro del problema, formando el dominio del problema.



- Problemas en clase:

Ejemplo 1.

14/sep/2022

Análisis del Problema

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

Restricciones: El número no puede ser cero

Datos entrada: Número real

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

Dominio: Todos los números reales.

- 1- Inicio
- 2- Determinar la variable
- 3- Pedir un número
- 4- Ingresar el número
- 5- Un número diferente a "0" (cero)
- 6- Si el número es "0", Regresar al Paso 3, si el número es $0 < N$, $N < 0$, pasa al siguiente paso.
- 7- Determinar si es positivo o negativo
- 8- Si es $N < 0$, $0 < N$ se acaba
- 9- Si no lo es, vuelve al paso 3/ 4.
- 10- Fin.

Iteración	X	salida
1	-5	El número es negativo
2	3	El número es positivo
3	-1	El número es negativo

Ejemplo 2.

14/sep/2022

Ejemplo 2

1 Inicio

2 Solicitar un número real (N1)

3 Solicitar un segundo número real (N2)

4 Si los números N1 y N2, se regresa al punto 3, a poner otro valor diferente $N1 = N2 = \text{paso 3}$

5 Si los números N1 y N2 son diferentes, se valida y avanza al siguiente paso

6 Se determina cual número es mayor, ya sea N1 o N2.

7 La salida determinará cual es mayor, si es N1 o N2.

8 Fin.

Iteración	N1	N2	Salida
1	4	8	El segundo número es mayor de los números
2	5	2	El primer número es mayor de los números
3	4	5	El segundo número es el mayor de los números

Ejemplo 3.

14/sep/2022

Ejemplo 3

1- INICIO

Factorial

$n! = n \times (n-1) \times \dots \times 1$

$4! = 4 \times 3 \times 2 \times 1 =$

Factorial de cero = 1

1- INICIO

2- Solicitar un número entero (x)

3- Si el número ingresado es menor a cero $x < 0$, regresar al paso 2

4- Si $x \geq 0$ se crea una variable entero contador, que iniciara en 2, y una variable factorial que inicie en 1.

5- Si $C \leq x$, es igual $C \cdot F = \text{Var } F$.

6- $C++1$

7- Regresar a 5

8- Fin

Iteración	x	Factorial	Contador
1	5	1	2
2	5	2	3
3	5	6	4
4	5	24	5
5	5	120	6

= Factorial es 120

Iteración	x	Factorial	Contador
1	2	1	2
2	2	2	3

Factorial = 2

Iteración	x	Factorial	Contador
1	4	1	2
2	4	2	3
3	4	6	4
4	4	24	5

Factorial = 24

Ejercicio en clase:

Ejercicio de Clase

- 1- INICIO
- 2- *Escriba "Escribe un número entero."
- 3- El usuario procede a escribir su número (VarNum)
- 4- Si $\text{VarNum} > 10$ se realiza $\text{VarNum} + 10 = (\text{VarResultado})$
- 5- *Escriba "El Resultado es:" (VarResultado)
- 6- Pasa al paso 9
- 7- En caso de que $\text{VarNum} \leq 10$, procede a $\text{VarNum} - 1 = (\text{VarResta})$
- 8- *Escribe "El resultado es" VarResta
- 9- Fin

Pruebas de Escrit.

Entrada	VarNum	Resultado	Salida
29	$29 + 10$	29	El número es mayor a 10
5	$5 - 1$	4	El número es menor a 10
11	$11 + 10$	21	El número es mayor a 10

- 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."

Una vez leído esto, un algoritmo es una sucesión de pasos a seguir para poder alcanzar un resultados a través de su proceso.

Debe cumplir algunas condiciones y/o características:

- 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

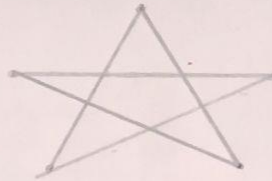
- Correcto: Cumplir con el objetivo.
- Debe tener al menos una salida y ésta debe de ser perceptible
- Debe ser sencillo y legible
- Eficiente: Realizarlo en el menor tiempo posible
- Eficaz: Que produzca el efecto esperado.

- Variables:

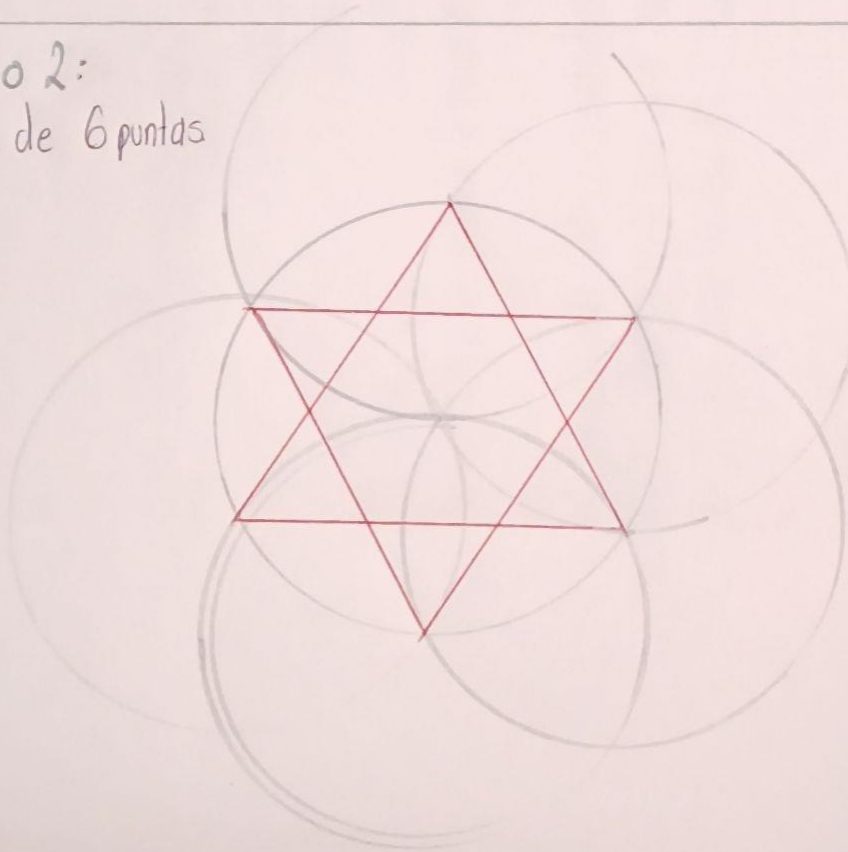
Las variables son caracteres que son permitirán guardar datos y valores, que dependerán de los datos de entrada, esto permite que el algoritmo pueda fluir en la secuencia de pasos a seguir.

- Ejercicio 1 y 2

Ejercicio 1: Estrella de 5 puntas



Ejercicio 2:
Estrella de 6 puntas



- Conclusiones:

Fue muy importante esta práctica por que, gracias a esta, ahora podemos saber como plantear mejor nuestros problemas a ciertos puntos de poder plasmarlos de la manera más fácil para poder explicarlo.

De igual manera también nosotros aprendimos a seguir nuestros pasos a tal punto que de esta forma nos es mas sencillo el poder practicar y poder resolver problemas que tal vez anteriormente se nos complicaba, así que el plantear un algoritmo es de suma importancia para poder resolver problemas y seguirlos al pie de la letra y después el poder comprobarlos.

- Referencias:

- Raghu Singh (1995). International Standard ISO/IEC 12207 Software Life Cycle Processes. Agosto 23 de 1996, de ISO/IEC. Consulta: Junio de 2015. Disponible en: <http://www.abelia.com/docs/12207cpt.pdf>
- Carlos Guadalupe (2013). Aseguramiento de la calidad del software (SQA). [Figura 1]. Consulta: Junio de 2015. Disponible en: <https://www.mindmeister.com/es/273953719/aseguramiento-de-la-calidad-del-software-sqa>
- Andrea S. (2014). Ingeniería de Software. [Figura 2]. Consulta: Junio de 2015. Disponible en: <http://ing-software-verano2014.blogspot.mx>
- Michael Littman. (2012). Intro to Algorithms: Social Network Analysis. Consulta Junio de 2015, de Udacity. Disponible en: <https://www.udacity.com/course/viewer#!/c-cs215/l-48747095/m-48691609>