|  |  |
| --- | --- |
| Descripción: escudofi_color_m2008_jpg | **Carátula para entrega de prácticas** |

Laboratorio de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Ing. Karina García Morales |
| *Asignatura:* | Fundamentos de Programación |
| *Grupo:* | **1122** |
| *No de Práctica(s):* | Practica de Laboratorio #3 Solución de  problemas y Algoritmos. |
| *Integrante(s):* | **Murua Norwood Cristian Giovanni** |
|  |  |
|  |  |
| *Semestre:* | 1er Semestre |
| *Fecha de entrega:* | 29 de agosto del 2017 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_  
  
  
**Guía práctica de estudio 03: 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.

**Actividades:**

A partir del enunciado de un problema, identificar el conjunto de entrada y el

conjunto de salida.

Elaborar un algoritmo que resuelva un problema determinado (dado por el

profesor), identificando los módulos de entrada, de procesamiento y de salida.   
  
  
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***: “La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software"(IEEE). 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 softwares. 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**

“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.”(ISO norma 12207)   
  
1. Definición de necesidades  
2. Análisis  
3. Diseño  
4. Codificación  
5. Pruebas  
6. Validación  
7. Mantenimiento y Evolución.  
 **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) 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.

Es importante identificar dos grandes conjuntos dentro del sistema: el conjunto de entrada y el conjunto de salida.   
*- 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.   
  
  
  
  
  
  
  
  
**Algoritmos**  
 Una vez realizado el análisis, se puede proceder al diseño de la solución, esto es, a la generación del 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. 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 esta debe de ser perceptible   
• Debe ser sencillo y legible   
• Eficiente: Realizarlo en el menor tiempo posible   
• Eficaz: Que produzca el efecto esperado

Las actividades a realizar en la elaboración de un algoritmo para obtener una solución a un problema de forma correcta y eficiente de un algoritmo.   
 **1. Resultados del análisis del problema**   
•Con qué datos se cuenta, cuáles son necesarios como valores de entrada, qué restricciones deben considerarse, cómo debe ser la salida para que el problema se resuelva.  
 **2. Construcción del algoritmo**   
•Se refiere a la descripción detallada de los pasos que deben seguirse para resolver el problema.   
 **3. Verificación del algoritmo**   
•Consiste en el seguimiento del mismo, empleando datos que son representativos del problema que se desea resolver (esto se conoce como prueba de escritorio).

**Un algoritmo consta de 3 módulos básicos:**

**Módulo de Entrada**

•Representa los datos que requieren para resolver el problema. Estos datos se

pueden solicitar al usuario, leer de un archivo, consultar de una base de datos, etc.

**Módulo de Procesamiento**

•Representa las operaciones necesarias para obtener un resultado a partir de los

datos de entrada.

**Módulo de Salida**

•Permite mostrar los resultados obtenidos a partir del módulo de

procesamiento de datos. Los resultados pueden mostrarse en diversos sitios: en la

pantalla, en un archivo, en una base de datos, etc.

**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 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.

  
\*¿Qué le falta al algoritmo?   
 Inicio y fin, para que fuese más definido pudiese haber tenido medidas exactas, pero la salida solo especifica una 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.

  
  
  
  
\*¿Qué me pareció?  
Me pareció muy completo porque cumple con casi todas las características del algoritmo:   
1. Preciso, indica todos los pasos sin ambigüedades.  
2. Definido, si realizo el mismo algoritmo dos veces me daría la salida que es una estrella de 6 puntos.  
3. Finito: no lo tiene ni el inicio. X  
4. Tiene salida (estrella de 6 puntos)   
5. Es muy sencillo, eficiente y eficaz.   
 **Ejercicio 3**  
Realiza el algoritmo para resolver una ecuación de segundo grado, identificando sus entradas, salidas y restricciones o soluciones.  
  
  
  
  
  
  
**Ejercicios propuestos:**

**1.** Relaciona el diagrama del ciclo de vida del software con las 5 tareas:

|  |  |
| --- | --- |
| **Tareas de la Ingeniería de Software** | **Ciclo del Software** |
| Planificación | Definición de necesidades |
| Análisis | Análisis |
| Diseño | Diseño |
| Codificación | Codificación |
| Verificación | Pruebas, validación, mantenimiento y evolución |

La Ingeniería del software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de los programas informáticos, uno de los métodos que usa es el ciclo del software, por lo tanto, este es una parte de la Ing. del software por eso son básicamente lo mismo las etapas que las tareas.  
  
  
**2**. Realiza el análisis, algoritmo y pruebas de escritorio para resolver el siguiente polinomio:  
  
3( 2x^3 -3x^2 +4x -2) = 6x^3 – 9x^2 + 12x – 6

Entrada: Dos polinomios  
Salida: Igualdad de polinomios  
Auxiliar: Leyes de los signos  
  
1. Inicio  
2. Multiplicar el primer término adentro del paréntesis que esta antes de la igualdad, por el 3 afuera del paréntesis, utilizando solo los coeficientes y las leyes de los signos.  
3. Multiplicar el segundo término adentro del paréntesis que esta antes de la igualdad, por el 3 afuera del paréntesis, utilizando solo los coeficientes y las leyes de los signos.  
 3.1 Después de sacar el producto del segundo término, sumárselo al resultado del paso 2, empleando las leyes de los signos.  
4. Multiplicar el tercer término adentro del paréntesis que esta antes de la igualdad, por el 3 afuera del paréntesis, utilizando solo los coeficientes y las leyes de los signos.  
   
 4.1 Después de sacar el producto del tercer término, sumárselo al resultado del paso 3.1, empleando las leyes de los signos.  
5. Multiplicar el cuarto término adentro del paréntesis que esta antes de la igualdad, por el 3 afuera del paréntesis, utilizando solo los coeficientes y las leyes de los signos.  
 5.1 Después de sacar el producto del cuarto término, sumárselo al resultado del paso 4.1, empleando las leyes de los signos.  
6. Has terminado, debes de tener una ecuación exactamente igual a la ecuación después del signo de igualdad.  
7. Fin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Iteración** | **X1** | **X2** | **Salida** |  |
| 1 | 2 | 1 | - | - |
| 2 | 1 | 1 | 1 | 1 |

Concluyo que es muy importante tener definido qué es un algoritmo, para qué se utiliza un algoritmo y cuáles son sus características, para después crear uno que sea eficiente y que nos ayude a resolver un problema. Los algoritmos los utilizamos desde la vida cotidiana como los es en el mercado para encontrar los artículos, hasta en nuestra próxima vida laboral en sistemas matemáticos para realizar una operación en una industria, por este último es de mi interés aprovechar la sabiduría de mi profesora para incrementar mi conocimiento en la elaboración de estos mismos.

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 delsoftware-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#!/ccs215/l-48747095/m-48691609