|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Karina Garcia Morales |
| *Asignatura:* | Fundamentos de Programación |
| *Grupo:* | 20 |
| *No. de práctica(s):* | 3 |
| *Integrante(s):* | Rosete Cruz Santiago |
| *No. de lista o brigada:* | 2 |
| *Semestre:* | 2 |
| *Fecha de entrega:* |  |
| *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.

**Conceptos**

**Problema informático:** Son las situaciones que, a cada una, le corresponde un conjunto de soluciones, además, guarda relación con cada instancia del problema con un subconjunto de solución.

**Solución de problemas:** Basándose en la definición de la IIEE, es aplicar, paso por paso, solución de manera rigurosa y cuantificable para desarrollar, operar y mantener el software.

**Ciclo de la vida del Software**: son todas las actividades que conllevan el desarrollo del software, desde su planeación y definición, hasta el fin de su uso. Se puede dividir en las siguientes etapas, que se siguen de manera cíclica

* Planeación
* Análisis de necesidades
* Diseño de estructuras de datos, es decir, donde se planea la arquitectura del programa y el funcionamiento del algoritmo para la solución al problema.
* Codificación
* Pruebas y validación, donde se valida que el software funciona y si puede solucionar el problema para el que fue diseñado.
* Evolución y mantenimiento, aquí se plantean cuestiones de cómo puede mejorar el software, donde se puede reiniciar el ciclo.

**Análisis:** Es la etapa del proceso donde se busca establecer cuáles son las necesidades del usuario e identificar con que elementos se posee para establecer una guía de trabajo, estos donde se poseen los fatos de entrada y que finalmente nos guían, mediante el algoritmo, a los datos de salida:

* **Datos de entrada:** los datos que son proporcionados a nuestro algoritmo o sistema.
* **Datos de salida:** son los datos que se emiten después de realizar los procesos indicados, que se obtienen gracias a los datos de entrada.

Es importante recalcar que la forma en la que se analiza el problema tiene que ser precisa y concisa, pues es la parte más importante del desarrollo del software, pues puede prestarse a ambigüedades que no tendrán soluciones o será diferente al problema planteado inicialmente.

**Dominio del problema**: es la unión del conjunto de entrada y de salida.

**Algoritmo:** son los pasos a realizar para llegar a un resultado o solución de un problema. La construcción del algoritmo requiere el establecimiento de reglas y el uso de un lenguaje específico, como una serie de pasos o procedimientos para lograr un resultado deseado. Es fundamental que el algoritmo sea preciso, finito, comprensible, sencillo de ejecutar, eficiente y eficaz, cumpliendo con el objetivo definido y con la capacidad de ser replicado para obtener resultados consistentes.

* Resultado del análisis de problema: Con que datos se cuenta para realizar alguna actividad, así como de que restricciones.
* Construcción del algoritmo: Elaboración de los pasos a seguir para resolver el problema
* Verificación: Una prueba realizada para saber si nuestro algoritmo funciona.

y la estructura del algoritmo es la siguiente:

* Entrada: Datos ingresados para que se lleve a cabo el procesamiento del algoritmo.
* Procesamiento: es donde se lleva a cabo los pasos y operaciones necesarias para el resultado
* Salida: muestra los resultados obtenidos en el procesamiento

**Desarrollo de la practica**

Se llevaron a cabo algunos ejemplos de algoritmos para resolver algunos problemas, y se propusieron soluciones y pasos que debe seguir el algoritmo para obtener una solución o resultado, llevada de manera matemática, es decir, que es un problema computable:

**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.

Algoritmo:

1.- Ingresar un número, ir al paso 2

2-  Si el número es 0, no se puede procesar, regresar al paso 1, pero si el numero x !=0, ir al paso 3.

3- Si el número x> 0, ir al paso 5, si x!>0, ir al paso 4.

4- Si el número x<0, ir al paso 6

5- Mostrar resultado “x es un número positivo”

6- Mostrar resultado “x es un número negativo”

Prueba de escritorio

|  |  |  |
| --- | --- | --- |
| Iteración | número (entrada) | salida |
| 1 | 1 | número positivo |
| 2 | 0 | ingresar nuevo número |
| 3 | -5 | número negativo |

**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.

Algoritmo:

Ingresar dos números: x,y, ir a paso 2

* si x=y, volver al primer paso, si no, paso 3
* Se le resta el número y al número x, (x-y), y si
  + el número resultante es >0, el número x>y, pasar al paso 4, de lo contrario, al paso 4.2
  + si el número resultante es <0 x<y, ir al paso 5
* mostrar resultado x>y “x es mayor que y”
* mostrar resultado x<y “x es menor que y”

Prueba de escritorio

|  |  |  |  |
| --- | --- | --- | --- |
| iteración | x | y | salida |
| 1 | 56 | 35 | x>y |
| 2 | 2 | 4 | x<y |
| 3 | 3 | 3 | - |

**Ejemplo 3**

PROBLEMA: Obtener el factorial de un número dado.

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

DATOS DE ENTRADA: Número entero.

DATOS DE SALIDA: El factorial del número.

DOMINIO: Todos los números naturales y el cero.

1. Ingresar un número “x” entero y >=0, ir a l paso
2. si x<0, regresar al paso 1, si no, al paso 3
3. si x>0 y es un número entero, se siguen los siguientes criterios
   1. x=0 mostrar el resultado 1, si no, ir al paso 3.b
   2. x=1 mostrar resultado 1, si no, ir al paso 3.c
   3. si x>1, multiplicar x · x-1, si el resultado x-1 >1 pasar al siguiente paso, si x-1=1, pasar al paso 4, si no, pasar al paso 3.d
   4. si x-1>1, realizar la operación x(x-1)(x-2)...(x-n), donde n es un número entero positivo, hasta que  x-n=1, pasar al paso 4
4. mostrar el resultado como: x!=resultado de paso (a, b, c, d según sea el caso)

***Ejercicio 1***

PROBLEMA: Seguir el algoritmo para obtener una figura

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

SALIDA: Figura correcta.

Algoritmo

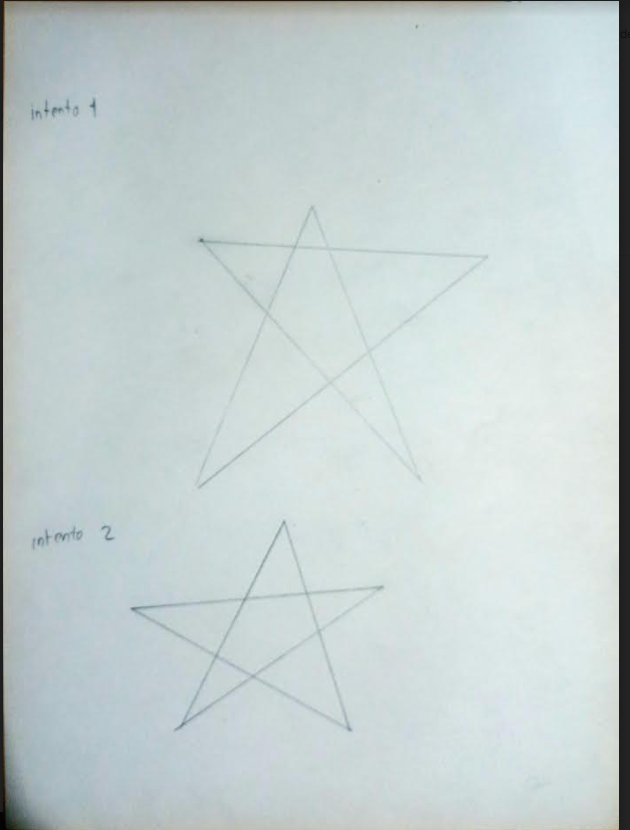
1. Dibuja una V invertida. Empieza desde el lado izquierdo, sube, y baja hacia el lado derecho, no levantes el lápiz.

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.



***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.

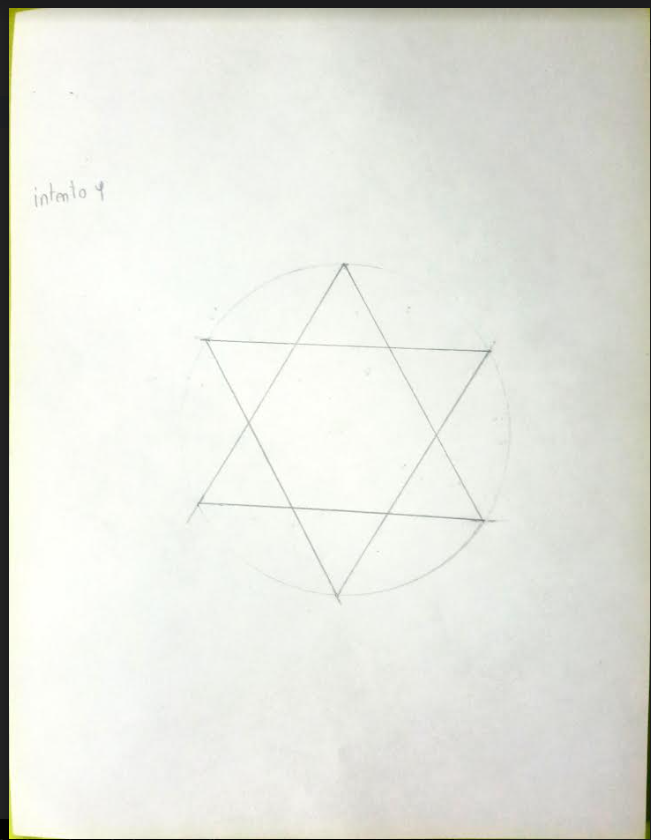
5. Ahora, coloca la punta del compás en uno de los puntos. Recuerda no cambiar el radio del compás. Haz otra marca en el círculo.

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.



**EJERCICIO 3**

* Genera un algoritmo para convertir grados Celsius a Fahrenheit

Problema: transformar grados Celsius a grados Fahrenheit

Restricciones: no puede ingresarse un número menor a -459.67

Dominio de entrada: Todos los numero reales mayores a -459.67

Dominio de salida: todos los números reales mayores a -459.67

Dominio [-459.67, ∞)

1 ingresar una cantidad x, e ir al paso 2

2 elegir grados Celsius (°C) o Fahrenheit (°F) ir al paso 2.1

2.1 si se elige °C, ir al paso 3, si no, ir al paso 2.2

2.2 si se elige °F, ir al paso 7

3 si x<-273.15, regresar al paso 2.1 e indicar (x=>-273.15), si no, ir al paso 4

4 Multiplicar el número x por 9/5, seguir al paso 5

5 sumarle 32 al resultado del paso 4, ir al paso 6

6 mostrar “x grados °C son igual a (resultado de paso 5) grados °F” ir al paso 11

7 si x<-459.67, mostrar x>-459.67 y regresar al paso 2.2, en caso contrario, seguir al paso 8

8 restar 32 al número x, ir al paso 9

9 multiplicar el resultado del paso 6 por (5/9), ir al paso 10

10 mostrar resultado “x grados °F son igual a “resultado de paso 9) grados °C”, ir a paso 11

11 mostrar opción de “nueva conversión”, ir al paso 1

**Actividades de tarea**

.

1.-Genera un algoritmo para resolver una ecuación de segundo grado.

Problema: Resolver ecuaciones de segundo grado

Restricciones: Debe existir un término cuadrático

Dominio de entrada: Números complejos

Dominio de salida: Números complejos

Dominio: Todos los números complejos

1.- Ingresar un número “a”, el termino cuadrático, ir al paso 1.1

1.1.-si a=0, volver al paso 1, si a! =0, ir al paso 2.

2.- Ingresar un número “b”, el término lineal, ir al paso 3.

3.- ingresar un número “c” el término independiente, ir al paso 4.

4.- obtener el numero inverso de b, ir al paso 5.

5.- obtener el cuadrado de b, ir al paso 6.

6.- realizar el producto de 4(a\*c), ir al paso 7.

7.- restar el resultado del paso 6 al resultado del paso 5, ir al paso 8.

8.- obtener la raíz cuadrada del paso 7, ir al paso 9.

9.- multiplicar el término a por 2 (2\*a), ir al paso 10.

10.- restar al paso 4 el resultado del paso 8, ir al paso 11.

11.- dividir el resultado del paso 10 entre el resultado del paso 9, ir al paso 12.

12.- sumar el resultado del paso 8 al resultado del paso 4, ir al paso 13.

13.- dividir el resultado del paso 12 sobre el resultado del paso 9, ir al paso 14.

14.- mostrar los resultados obtenidos en los pasos 11 y 13.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Prueba | a | b | c | operaciones |
| 1 | 5 | 4 | 2 | -5 |
|  | 5 | 4 | 2 | 25 |
|  | 5 | 4 | 2 | 40 |
|  | 5 | 4 | 2 | 25-40=-15 |
|  | 5 | 4 | 2 |  |
|  | 5 | 4 | 2 | 8 |
|  | 5 | 4 | 2 | -5- |
|  | 5 | 4 | 2 |  |
|  | 5 | 4 | 2 |  |
|  | 5 | 4 | 2 |  |
|  | 5 | 4 | 2 | y |

**Conclusiones**

Para resolver problemas computables en la programación, resulta indispensable analizar correctamente un problema para obtener ideas sobre posibles soluciones que se adecuen al contexto del planteamiento inicial, para así dar una solución eficiente y eficaz al problema.

Al momento de buscar la mejor solución, es importante destacar que los procedimientos para llegar a la misma, deben ser claros y llevar orden, así como hacer pruebas para verificar que nuestro planteamiento es correcto y da solución al problema dado, de forma que si falla, se pueda buscar el error y repararlo o mejorarlo para poder llegar a un algoritmo que nos permita resolver cualquier problema con datos de entrada similares al planteado inicialmente.

**Bibliografía:**

Solano Gálvez, J. A., García Cano, E. E., Sandoval Montaño, L., Quezada Reyes, C., Arteaga Ricci, T. I., Morales Nava, M. G., . . . Zúñiga Barragán, H. (21 de febrero de 2022). *Manual de Prácticas*, 04. Recuperado el 19 de febrero de 2024, de Facultad de Ingenieria: http://lcp02.fi-b.unam.mx