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

Laboratorios de computación

salas A y B

|  |  |  |
| --- | --- | --- |
| *Profesor:* | Ing. Claudia Rodríguez Espino | |
| *Asignatura:* | Fundamentos de Programación | |
| *Grupo:* | 1104 | |
| *No de Práctica(s):* | 03 | |
| *Integrante(s):* | Peñaloza Lugo Tania Lizeth | |
| *Semestre:* | 4° | |
| No. de Equipo de cómputo empleado: | | | 58 |
| *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.

**Actividades:**

Una vez leído la introducción se pudieron definir y reconocer los métodos para generar un software así como el ciclo de vida del mismo. La primera parte para la elaboración de un software es el análisis del problema y con ello identificar los datos de entrada, proceso y salida para poder desarrollar el algoritmo. De esta forma se realizaron 3 actividades en las cuales se identificaron los datos de entrada, el proceso y los datos de salida, acompañados de su algoritmo y una prueba de escritorio para corroborar su eficacia.

1. **Área de un círculo**

**Datos de entrada:** radio de círculo

**Proceso:** pi = 3.1416

A = r\*r\*pi

**Datos de salida:** A

**Algoritmo**

1. Inicio
2. Pi= 3.1416
3. Solicitar el radio del circulo
4. A= r\*r\*pi
5. Mostrar A
6. fin

**Pruebas de escritorio**

|  |  |  |
| --- | --- | --- |
| **Iteración** | **r** | **salida** |
| 1 | 2 | 6.2831 |
| 2 | 5 | 78.54 |
| 3 | 7 | 153.93 |

1. **Y >2 y = x2 +3x – 2**

**Y<2 y= 2x2 +x + 3**

**Datos de entrada**: numero x

**Proceso:** si x > 2 resolver para 1, sino resolver para 2

**Datos de salida:** mostrar y

**Algoritmo:**

1. Inicio
2. Solicitar valor de x
3. Si x >2 entonces
   1. Y = x^2 +3x – 2 e ir a paso 7
4. Si x<2 entonces
   1. y= 2x^2 –x +3 e ir a paso 7
5. Si no, ir a paso 6
6. No hay solución
7. Mostrar y
8. Fin

**Pruebas de escritorio:**

|  |  |  |
| --- | --- | --- |
| **Iteración** | **X** | **salida** |
| 1 | 2 | No hay solución |
| 2 | 5 | 38 |
| 3 | 0 | 3 |

1. **Resolver para cualquier valor de x en la fórmula general**

**Datos de entrada:** a b y c

**Proceso:**

n= raíz de b cuadrada menos 4ac

si n < 0

multiplicar por menos 1 y sumar i

sino

z= -b más menos n

d= 2a

x = z entre d

**Datos de salida:** x1 y x2

**Algoritmo:**

1. Inicio
2. Solicitar el valor para a
3. Solicitar el valor para b
4. Solicitar el valor para c
5. Si a = 0 regresar a paso 2
6. d= 2a
7. si n < 0

multiplicar por menos 1

r = raíz de n + i

1. si n >0

r = raíz de n

1. x1 = z1 entre d
2. x2 = z2 entre d
3. mostrar x1 y x2
4. fin

**Prueba de escritorio**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Iteración** | **a** | **b** | **c** | **X1** | **X2** |
| 1 | 0 | 1 | 2 | ------- | ------ |
| 2 | 5 | 2 | 1 | 0+i | 1/5+i |
| 3 | 1 | 4 | 1 | -0.26 | -3.73 |
| 4 | 4 | 5 | 2 | 1.17+i | 1.52+i |

**Conclusión**

Saber analizar un problema para así llegar a la mejor solución es el proceso más complicado de realizar para poder elaborar un software eficiente, sin embargo al seguir de forma ordenada y lógica los procesos que se necesita llevar a cabo la construcción del mismo, se ve reflejado en el algoritmo y por esta razón considero que es la parte más importante al programar.