|  |  |  |
| --- | --- | --- |
|  | **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 programacion |
| *Grupo:* | 20 |
| *No de Práctica(s):* | Practica 3 Guía práctica de estudio 03: Solución de problemas y Algoritmos. |
| *Integrante(s):* | Edgar Humberto Del Campo Valdés |
| *No. de Equipo de cómputo empleado:* | N/A |
| *No. de Lista o Brigada:* | 10 |
| *Semestre:* | 2021-1 |
| *Fecha de entrega:* | Jueves 29 de Octubre de 2020, 19:00 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Practica 3 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. (Solano, 2020)

**Desarrollo de practica**

Ejemplo 1

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

DATOS DE ENTRADA: Todos los números reales

RESTRICCIONES: El número no puede ser cero.

DATOS DE SALIDA: Saber si los números reales son positivos o negativos

Ejemplo 2

PROBLEMA: Obtener el mayor de dos números diferentes dados.

RESTRICCIONES: Los números no pueden ser iguales

Datos de entrada: Todos los números reales N1≠N2

Datos de salida: La obtención del numero mayor de dos números reales

**Ejercicios de tarea**

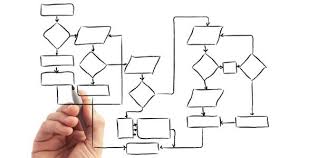
1. ¿Qué se necesita para dar solución a un problema?

Para resolver un problema es necesario aplicar un proceso en el que puedas organizar las etapas para llevar acabo la solución de este. La Ingeniería de Software provee métodos que indican cómo generar software. 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).

2. Describe las fases del ciclo de vida del desarrollo de software e ilustralas



3. Analizar las siguientes problemáticas, desglosando y dando solución al problema identificando claramente su conjunto de entrada (datos de entrada), el conjunto de salida (datos de salida) y restricciones, si es que tiene.

* En una playa el estacionamiento cobra $ 2.5 por hora o fracción. Determinar cuánto debe pagar un cliente por el estacionamiento de su vehículo, conociendo el tiempo de estacionamiento en horas y minutos.

Datos de entrada: un estacionamiento cobra $2.5 por hora

Datos auxiliares: por cada minuto se cobra $0.041

Restricciones: no hay tiempo de tolerancia, se cobra desde el primer minuto

Datos de salida: el monto a pagar por el tiempo que estuvo en el estacionamiento

* Datos dos números realizar suma, resta, producto y división (considerar el dividendo ≠ 0).

Datos de entrada: 2 números del conjunto de los reales, operaciones a realizar (suma, resta. Producto, y división)

Restricciones: dividendo ≠ 0

Datos de salida: los resultados de las operaciones de dos números (conjunto de los reales)

* Dados tres números; si el primero es negativo, deberá realizar el producto de los tres, y en caso contrario realizará la suma.

Datos de entrada: tres números del conjunto de los reales

Restricciones: si el primero es negativo, deberá realizar el producto de los tres, y en caso contrario realizará la suma.

Datos de salida: el resultado de las operaciones correspondientes (conjunto de los números correspondientes

**Conclusión**

Me es bastante interesante saber la estructura que tienes que seguir para desarrollar un software que no solamente es el programar y ya, que conlleva todo un proceso que va desde escribir los datos de entrada y va hasta su mantenimiento, ya que para hacer con éxito el programa tiene que pasar por varios filtros, varias pruebas, y entre mas organizado este el proceso mejor va a quedar el resultado, es como un instructivo muy detallado y que no tiene que llevar errores, ya que la computadora es simple, lo que le metas es lo que hace, si tu análisis esta mal por lo tanto la programación va a estar aun peor y te vas a tener que regresar, creo que es muy importante tener las bases claras para poder hacer algo mas grande y complejo

# Bibliografía

Solano, I. J. (2020). *tuaulavirtual*. Obtenido de https://tuaulavirtual.educatic.unam.mx/pluginfile.php/696595/mod\_resource/content/2/MADO-17\_FP.pdf