



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación  
Salas A y B

*Profesor:* Alejandro Pimentel

*Asignatura:* Laboratorio de programación

*Grupo:* 135

*No de Práctica(s):* Práctica 1

*Integrante(s):* Pimentel Maldonado Adriana Itzel

*No. de Equipo de cómputo  
empleado:* 13

*No. de Lista o Brigada:* 42

*Semestre:* Primer semestre

*Fecha de entrega:* 02/septiembre/2019

*Observaciones:*

CALIFICACIÓN: \_\_\_\_\_

**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

## INTRODUCCIÓN

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 que de acuerdo a la IEEE se define como "La aplicación de un enfoque sistemático, disciplinado y cuantificable hacia el desarrollo, operación y mantenimiento del software". 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 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).

### *Ciclo de vida del software*

La ISO 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***

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).

Por lo tanto, la etapa del análisis consiste en conocer qué es lo que está solicitando el usuario. Para ello 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 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.

## ***Algoritmos***

Una vez realizado el análisis, es decir, ya que se entendió qué es lo que está solicitando el usuario y ya identificado el conjunto de entrada y el conjunto de salida, 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.

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**

Un algoritmo consta de 3 módulos básicos:

1. **Módulo de entrada:** representa los datos que se requieren para resolver el problema
2. **Módulo de procesamiento:** Representa las operaciones necesarias para obtener un resultado a partir de los datos de entrada
3. **Módulo de salida:** Permite mostrar los resultados obtenidos a partir del módulo de procesamiento de datos.

### **ACTIVIDAD 1.**

Explicar las precondiciones y el conjunto de los algoritmos para:

a) Problema: Pescar

Precondiciones:

- ✓ Disponer de una caña de pescar, anzuelos, carnada y un contenedor.
- ✓ Encontrarse en alguna laguna, lago, río, o cualquier cuerpo de agua que pueda contener el objetivo deseado

Datos de entrada: Caña con anzuelo

Datos de salida: Captura del pez

Solución del problema:

1. Colocar carnada en el anzuelo.
2. Lanzar el anzuelo al cuerpo de agua en cuestión.
3. Esperar unos minutos, o hasta que la cuerda sea tensada.
4. Tirar del carrete e ir soltando en repetidas ocasiones.
5. Deposita al pez en un contenedor.

**b) Problema: Lavarse las manos**

Precondiciones:

- ✓ Tener las manos sucias, o la necesidad de lavarlas.
- ✓ Disponer de agua, jabón y toalla.

Datos de entrada: Manos sucias

Datos de salida: Manos limpias

Solución:

1. Colocarse cerca del lavamanos o toma de agua que se disponga.
2. Mojar cada una de las manos durante unos segundos.

3. Con ayuda del jabón, lavar la superficie de los dedos, palma y cara posterior.
4. Sumergir las manos nuevamente en agua y retirar el jabón.
5. Secar las manos con la toalla.

**c) Problema: Cambiar una llanta.**

Precondiciones:

- ✓ Tener un auto con llanta dañada o desgastada.
- ✓ Disponer de herramientas como gato hidráulico y llave.
- ✓ Tener una llanta nueva

Datos de entrada: Carro con llanta dañada

Datos de salida: Carro con llantas óptimas

Solución:

1. Estar estacionado.
2. Aflojar las tuercas del coche con la llave
3. Con ayuda del gato hidráulico levantar el auto.
4. Retirar la llanta dañada.
5. Colocar la llanta nueva.
6. Colocar de vuelta a su lugar las tuercas.
7. Quitar el gato hidráulico.

**ACTIVIDAD 2.**

Desarrollar los algoritmos para:

**a) Problema: Determinar si un número es positivo o negativo.**

Precondiciones:

- ✓ Debe ser un número real
- ✓ Debe ser diferente de cero

Datos de entrada: Un número entero

Datos de salida: Verificar si el número es positivo o negativo.

Solución del problema:

1. Dar entrada al número real en cuestión
2. De ser cero el número, regresar al paso 1
3. Cuando el número que añadimos es diferente y mayor a cero, este es positivo
4. Cuando el número que añadimos es diferente y menor que cero, este es negativo.

**b) Problema: Obtener el mayor de dos números diferentes**

Precondiciones:

- ✓ Dos números enteros diferentes entre sí

Datos de entrada: Dos números enteros.

Datos de salida: Verificar qué número es el mayor.

Solución:

1. Ingresar al sistema el primer número entero.
2. De no ser un número entero, regresar al paso uno.
3. Ingresar al sistema el segundo número entero.
4. De no ser un número entero y diferente del primero, regresar al paso tres.
5. Si el primer número entero es más grande que el segundo, se confirma que es el mayor de ambos.
6. Si el segundo número entero es más grande que el primero, se confirma que es el mayor de ambos.

**c) Problema: Obtener la factorial de un número.**

Precondiciones:

- ✓ El número ingresado debe ser un entero, diferente de cero y de valor positivo

Datos de entrada: Número entero

Datos de salida: La factorial del número entero.

Solución:

1. Ingresar al sistema un número entero
2. De ser el número igual o menor a cero, volver al paso 1
3. Multiplicar el número por cada uno de los números que le precede que sean enteros, positivos y diferentes de cero.

**ACTIVIDAD 3.**

Verificar sus algoritmos anteriores, al “ejecutarlos” paso a paso con los siguientes valores.

**Entrada**

**Solución**

**Salida**

54	$54 > 0$	Número positivo
-9	$-9 < 0$	Número negativo
-14	$-14 < 0$	Número negativo
8	$8 > 0$	Número positivo
0	$0 = 0$	0 (Error)

Entrada	Solución	Salida
(4,5)	$4 < 5$	5 es el mayor
(-9,16)	$-9 < 16$	16 es el mayor
(127.8+4i)	$127.8+4i$	127.8+4i (Error)
(7,m)	7,m	7,m (Error)

Entrada	Solución	Salida
5	5(4)(3)(2)(1)	120
9	9(8)(7)(6)(5)(4)(3)(2)(1)	362880
0	0	0 (Error)
-3	-3	-3 (Error)

#### ACTIVIDAD 4.

Desarrollar algoritmos propios de un procesador (asignando registros genéricos) para:

##### a) Cambiar el signo de un número binario.

Precondiciones: Tener un número en base binaria.

Solución:

1. Escribir el número deseado.
2. Ir transcribiendo el número normalmente, empezando por el último dígito a la derecha.
3. Dejar de transcribir cuando lleguemos al primer "1" de la cifra, este será el último dígito que pasará tal cual.
4. Invertir los números posteriores al primer "1", es decir, al encontrarse un "0" se cambiará por un "1" y viceversa.
5. Invertir también el último dígito (Bit de signo) para indicar si es un número positivo (0) o negativo (1).

##### b) Hacer una suma larga binaria.

Precondiciones: Tener un número en base binaria.

### Solución:

1. Se acomodan los números, uno debajo del otro, asegurándonos de posicionarlos de acuerdo a su valor (unidad, decena, etc)
2. Ubicarse en la primera columna del extremo derecho y se suman los dígitos de cada columna. Este paso se repite tantas veces como columnas se tenga, se avanza de derecha a izquierda.
3. Cuando la suma de una columna sobre pasa el número 1, se coloca un cero como resultado y se pasa el 1 a la siguiente. Este paso se repite tantas veces como sea necesario.

### **CONCLUSIONES:**

El uso de los algoritmos nos ayuda a llevar a cabo la solución de un problema, pues nos da los pasos a seguir para llegar a dicha respuesta en caso que las precondiciones estén cumplidas, y el que nuestro sistema no pueda hallarla, no quiere decir que está mal, sino que alguno de los requisitos no fue cumplido satisfactoriamente.

Por otro lado, es de suma importancia el planteamiento y asignación de las variables, pues en este paso se definirá si verdaderamente el algoritmo que creamos son registros genéricos que se pueden aplicar a cualquier situación planteada, y no solo a algunos casos particulares.