

	<b>Carátula para entrega de prácticas</b>	
Facultad de Ingeniería	Laboratorio de docencia	

## Laboratorios de computación salas A y B

*Profesor:* Manuel Enrique Castañeda Castañeda

*Asignatura:* Fundamentos de Programación

*Grupo:* 14

*No. de Práctica(s):* 3

*Integrante(s):* Aburto López Roberto, Aguilar Pérez José  
Ramón, Cruz Soria Oscar, Téllez Cruz  
Brayan, Zurita Cámara Juan Pablo

*No. de Equipo de*  
*cómputo empleado:* \_\_\_\_\_

*No. de Lista o Brigada:* Brigada 12

*Semestre:* Primer semestre

*Fecha de entrega:* 24 de septiembre del 2021

*Observaciones:* \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

**CALIFICACIÓN:** \_\_\_\_\_

# Cuestionarios Previos

Cruz Soria Oscar Gpo 14

## Cuestionario Previo Práctica 3

### 1: ¿Qué es un algoritmo?

Un algoritmo nos permite llevar a cabo una tarea o encontrar la solución a un determinado problema a través de un flujo de **instrucciones bien definidas y estructuradas** que deben estar ordenadas y ser finitas, osea que deben tener solución.

Permiten definir una serie de pasos y reglas definidas para resolver un problema.

### 2: Describa las características de los algoritmos

Los puntos fundamentales de un algoritmo son:

- Un algoritmo debe ser **preciso**, es decir, debe indicar el orden exacto de ejecución de cada paso implicada
- Un algoritmo debe estar definido, siempre se debe obtener el mismo resultado
- Un algoritmo debe ser finito, debe tener un numero de pasos hasta finalizar la tarea
- Un algoritmo debe ser legible, el texto debe ser claro y conciso, la comprensión del algoritmo debe ser inmediata.



## Cruz Soria Oscar

- Un algoritmo debe tener Entrada, Proceso y Salida
- Un algoritmo debe ser correcto, no presenta errores
- Un algoritmo debe ser eficaz
- Un algoritmo debe tener solución.

### 3) ¿Qué es un pseudocódigo?

Se basa en el lenguaje de programación real, donde se expresan las instrucciones en un lenguaje común para facilitar la escritura y lectura.

La finalidad es representar la solución y proceso a un algoritmo de la forma más sencilla y detallada. Es un lenguaje simplificado entre el programador y la máquina.

### 4: ¿Cuáles son las etapas del desarrollo de software? Ciclo infinito ↻

1) Definición de necesidades

2) Análisis

3) Diseño

4) Codificación

5) Pruebas

6) Validación

7) Mantenimiento y evolución

Zurita Cámara Juan Pablo  
Cuestionario Previo

### Práctica 3: Solución de problemas y Algoritmos

#### 1. ¿Qué es un algoritmo?

Es un método para resolver problemas mediante una serie de pasos definidos, finitos y precisos.

#### 2. Describe las características de los algoritmos.

**Precisos:** Tienen que indicar el orden de realización de cada paso. No debe presentar ambigüedad.

**Definido:** Se debe obtener el mismo resultado cada vez.

**Finito:** Debe de tener un inicio y fin.

**Correcto** <sup>Se puede</sup>  
~~con lápiz y hoja.~~ resolver

**Efícaz:** Cada instrucción se debe realizar a papel y lápiz.

**Salida:** Nos muestra la solución.

**Entrada:** Se le introducen datos.

**Solución:** Debe tener una solución.

#### 3. ¿Qué es el pseudocódigo?

Forma de expresar los distintos pasos que va a realizar un programa de la forma más cercana al lenguaje de programación que vayamos a utilizar. Es un código escrito para que lo lea/entienda un ser humano y no un ordenador.



Zurita Gómara Juan Pablo

4. ¿cuáles son las etapas del desarrollo de un software?

1 Definición de necesidades

2 Análisis

3 Diseño

4 Codificación

5 Pruebas

6 Validación

7 Mantenimiento y evolución.

Roberto Alberto López

## 1. ¿Qué es un algoritmo?

Un algoritmo es un método para resolver problemas mediante series de pasos precisos, definidos y finitos. Es una serie de operaciones detalladas que pueden formularse de muchas formas, siempre y cuando no haya ambigüedades.

## 2. Describe las características de los algoritmos.

- **Preciso:** Debe estar en orden los pasos y no deben presentar ambigüedades.
- **Definido:** Deben obtener el mismo resultado.
- **Finito:** El algoritmo debe tener un final.
- **Correcto:** No debe tener errores.
- **Entrada:** Un algoritmo tiene de 0 o más entradas, cantidades que le son dadas antes de que comience el algoritmo, o mientras el algoritmo corre. Estas entradas son tomadas de conjuntos específicos de datos.
- **Salida:** Un algoritmo tiene una o más salidas, estas cantidades tienen relación con las entradas.
- **Eficacia:** Pueden ser hechos de forma exacta y en tiempo finito por un humano usando lápiz y papel.
- **Solución:** Debe llegar a un resultado.



Roberto Aburto López

## 3. ¿Qué es pseudocódigo?

Es una forma de escribir los pasos que va a realizar un programa de la forma más cercana al lenguaje de programación que vamos a utilizar posteriormente.

## 4. ¿Cuáles son las etapas del desarrollo de software?

- **Planificación:** planificar qué función tendrá el software.
- **Análisis:** averiguar qué exactamente lo que tiene que hacer el software.
- **Diseño:** Decidir la estructura general del software.
- **Implementación:** elegir las herramientas adecuadas que facilite el trabajo y un lenguaje de programación adecuado para el software a construir.
- **Pruebas:** Detectar los fallos que pueda tener el software.
- **Instalación o despliegue:** Poner el software en funcionamiento.
- **Uso y mantenimiento:** Detectar los errores obtenidos con el tiempo y actualizarlo a las necesidades actuales.

## Cuestionario Previo : Practica 3 : Solución

### ¿Qué es un algoritmo?

Es un conjunto de pasos ordenados que buscan resolver un problema, son instrucciones que siguen una secuencia.

### Describe las características de los algoritmos:

- Es finito: Debe acabar en algún momento.
- Puede ser cualitativo o cuantitativo, es decir describir procesos textuales o numéricos.
- Es preciso: Realizar los pasos es posible sin hacer nada extra.
- Su resultado es exacto.
- Puede tener una o más entradas de datos.
- Debe tener al menos una salida del resultado.
- Debe llegar a una solución.

### ¿Qué es el pseudocódigo?

Es la forma de representar un algoritmo en alto nivel codando las instrucciones descritas de forma ordenada, nos permite tener una vista previa del código.

Se usa generalmente para iniciarse en la programación.

### ¿Cuáles son las etapas del desarrollo de software?

- 1.- Búsqueda o presentación del problema
- 2.- Análisis del problema
- 3.- Propuesta de solución
- 4.- Planificación
- 5.- Codificación del sistema
- 6.- Pruebas constantes
- 7.- Depuración
- 8.- Lanzamiento
- 9.- Mantenimiento y mejoras



## **Guía práctica de estudio 03: 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.

### **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.

### **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 con 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).

## Desarrollo de la práctica: ALGORITMOS

### 1. Problema: Encontrar las raíces de un polinomio de segundo grado.

**Datos de entrada:** Números reales para formar el polinomio.

**Datos de salida:** Las raíces del polinomio, que son números reales.

**Dominio:** Números reales.

**Análisis del problema:** Una ecuación de segundo grado se define de la siguiente forma :  $ax^2 + bx + c = 0$  .

### Solución

#### 1. Formar el polinomio.

- Solicitar el valor  $a$  de la forma  $ax^2 + bx + c = 0$ , y guardar la variable.
- Solicitar el valor  $b$  de la forma  $ax^2 + bx + c = 0$ , y guardar la variable.
- Solicitar el valor  $c$  de la forma  $ax^2 + bx + c = 0$ , y guardar la variable.

**Nota:**  $a$ ,  $b$  y  $c$  deben de ser números reales.

#### 2. Ya formado el polinomio, formar con las variables:

$$-b \pm \sqrt{b^2 - 4ac} \div 2a$$

**Nota:** Checar que la parte de  $\sqrt{b^2 - 4ac}$  sea  $\geq 0$ . Si es menor a cero pedir otro valor para  $c$ .

- Resolver  $x = -b + \sqrt{b^2 - 4ac} \div 2a$ , y nombrar al resultado como  $x_1$
- Resolver  $x = -b - \sqrt{b^2 - 4ac} \div 2a$  y nombrar al resultado como  $x_2$
- Mostrar el resultado de  $x_1$  y  $x_2$ , estas son las raíces del polinomio.



## 2. Problema: Determinar si un número es primo

**Datos de Entrada:** Números naturales.

**Datos Salida:** Es primo o no es primo.

### Solución

1. Definir contador Divisiones Válidas.
2. Solicitar un número al usuario ( N ).
3. Crear un contador "i" que inicia en 1.
4. Dividir el número entre el "i", Si el residuo del número ingresado sobre el contador es 0, entonces aumenta uno al contador de divisiones válidas.
5. Sumar 1 y volver al paso 4 hasta que el "i" sea igual al valor de N.
6. Si el contador de divisiones válidas es mayor a 2, indicarle al usuario que su número no es primo, si es menor o igual a 2 indicarle al usuario que su número si es primo.
7. Fin.

### 3. Problema: Obtener el promedio de calificaciones.

**Restricciones:** Ingresar calificaciones negativas o mayores a 10.

**Datos de entrada:** Las calificaciones y el número de calificaciones.

**Datos de salida:** El promedio de calificaciones.

**Dominio:** 0 y los números naturales hasta el 10.

#### Solución

1. Solicitar el número de calificaciones, y guardarlo en una variable "**n**".
2. Establecer una variable como "**suma**"=0. Establecer una variable "**i**"=0.
3. Solicitar calificación, guardarla como "**x**".
4. Establecer que "**suma**"="**suma**"+"**x**". Establecer "**i**"="**i**" +1
5. Repetir los pasos 3 y 4 hasta que "**i**"=n
6. Realizar "**suma**"/"**n**", este resultado es el promedio.
7. Mostrar el promedio.



#### 4. Problema: Sumar o restar dos números complejos, que determine el profe.

**Restricciones:** Ninguna

**Datos de entrada:** Conjunto de los números complejos.

**Datos de salida:** Conjunto de los números complejos.

**Dominio:** Conjunto de los números complejos.

#### **Análisis del problema:**

Recordar que un número complejo se define como un número de la forma  $a + bi$ , donde  $a$  y  $b$  son números reales e  $i$  es la unidad imaginaria.

#### **Solución:**

1. Preguntar si es suma o resta.
2. Formar el primer número complejo. Solicitar los valores para  $a$  y  $b$  para cumplir con forma  $a + bi$ .
3. Formar el segundo número complejo. Solicitar los valores para  $a$  y  $b$  para cumplir con forma  $a + bi$ .

**Nota:**  $a$  y  $b$  solo pueden ser números reales.

4. Para sumar o restar dos números complejos, se suman o restan separadamente las partes reales e imaginarias, del mismo modo como se reducen los términos semejantes.

#### **Ejemplo:**

$$(3 + 2i) + (-2 + 5i) = 3 + 2i - 2 + 5i = 1 + 7i$$

**Nota:** En caso de que se busque realizar una resta, el primer número complejo formado será el minuendo y el segundo número complejo formado será el sustraendo.

5. Mostrar el resultado ya sea de la suma o de la resta.

## 5. Problema: Ingresar tres números y determinar cuál es el mayor

**Restricciones:** Para ejecutar este programa todos los números deben de ser diferentes y deben de ser números reales.

**Datos de entrada:** Números reales

**Datos de salida:** El número mayor

**Dominio:** Todos los números reales.

### Solución:

1. Solicitar el primer número real, se almacena en “a”.
2. Solicitar el segundo número real, se almacena en “b”.
3. Solicitar el tercer número real, se almacena en “c”.
4. Si  $a > b$  (a es mayor a b) y  $a > c$  (a es mayor a c), entonces **Escribir “El número “a” es el mayor de los tres”**.
5. Si,  $b > c$  y  $b > a$ , entonces **Escribir “El número “b” es el mayor de los tres**.
6. Si  $c > a$  y  $c > b$ , entonces **Escribir “El número “c” es el mayor de los tres**.



**6.- Problema:** Ingresar tres números y determinar cuál es el menor o si son iguales.

**Restricciones :** Los números deben ser reales.

**Datos de entradas :** Números reales

**Datos de salidas :** Decirnos el número mayor o si los números son iguales.

### Solución

1. Solicitar un número real, y guardarlo como A.
2. Solicitar un número real, y guardarlo como B.
3. Solicitar un número real, y guardarlo como C.
4. Checar si  $A=B$ .
  - a. Si es cierto: Checar si  $B=C$ .
    - i. Si es cierto: Decirle al usuario que todos los números son iguales.
    - ii. Si no es cierto: Checar si  $A<C$ .
      1. Si es cierto: Decirle al usuario que "A y B son los números menores".
      2. Si no es cierto: Decirle al usuario que "C es el número menor."
  - b. Si no es cierto: Checar si  $A=C$ 
    - i. Si es cierto: Checar si  $C<B$ 
      1. Si es cierto: Decirle al usuario que "A y C son los números menores."
      2. Si no es cierto: Decirle al usuario que "B es el número menor."
    - ii. Si no es cierto: Checar si  $A<B$ 
      1. Si es cierto: Checar si  $A<C$ 
        - a. Si es cierto: Decirle al usuario que "A es el menor"
        - b. Si no es cierto: Checar  $B<C$ 
          - i. Si es cierto: Decirle al usuario que "B es el menor".
          - ii. Si no es cierto: Decirle al usuario que "C " es el menor."

**7.- Problema:** Ingresar dos números a y b, determinar si a es mayor obtener la multiplicación de  $a*b$ , si b es mayor obtener la división  $a/b$ .

**Restricciones :** Ninguna

**Datos de Entradas :** Números reales.

**Datos de Salida :** Números reales.

1. Solicitar un número real y guardarlo como "a".
2. Solicitar otro número real y guardarlo como "b".
3. Si "a" es mayor que "b", realizar el producto  $a*b$ . Mostrar el resultado.
4. Si "b" es mayor que "a":
  - a. Si "b" es 0, entonces mostrar que no es una división válida.
  - b. Si "b" es diferente a 0 **realizar  $a/b$** . Mostrar el resultado.
5. Si "a" y "b" son iguales, imprimir "Los números A y B son iguales",
6. Fin

**8.- Problema:** Obtener 2 a la  $n$  potencia,  $n$  lo determina el usuario

**Restricciones:** “ $n$ ” no puede ser un número complejo.

**Datos de entrada:** Los datos de entrada van hacer para el valor de  $n$ , y van a ser los números reales.

**Datos de salida:** Los datos de salida nos van a dar el resultado de la potencia, y van a ser los números reales.

**Dominio:** Los números reales.

**Solución:**

1. Solicitar un número real y almacenarlo en una variable (recomendamos que se llame “ $n$ ”).
2. Elevar 2 a la potencia “ $n$ ”, no obstante, si “ $n$ ” es negativo, elevar 2 a la potencia “ $n$ ” y que ese resultado divida a 1.
3. Mostrar el resultado.



## **OBSERVACIONES**

### **Cruz Soria Oscar.**

No hubo mayor problema al realizar la práctica, hubo organización desde un principio y se repartieron los ejercicios asignados, de manera que todos aportaran algo en la misma, al final todos la revisamos para comprobar que no faltase algo.

### **Zurita Cámara Juan Pablo**

Fue una práctica relativamente sencilla debido a que ya hemos estado trabajando ejercicios similares con el profesor.

### **Aburto López Roberto**

No hubo tanta complicación al realizar esta práctica, debido a que alguno de los algoritmos ya los habíamos trabajado en clase, además que es mucho más fácil hacer los pseudocódigos debido a que es a nuestro idioma.

### **Téllez Cruz Brayan**

La mayoría de los problemas que tuvimos que realizar fueron sencillos y no nos tomaron grandes dificultades, salvo por ejemplo la suma de imaginarios, pues primero tuvimos que investigar como poder hacerlo antes de realizar su algoritmo. De ahí en fuera considero que se nos hicieron sencillos de llevar y resolver.

## **CONCLUSIONES**

### **Cruz Soria Oscar.**

Se realizaron de manera satisfactoria cada uno de los ejercicios propuestos, donde se elaboró de manera correcta cada uno de los algoritmos siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software, donde se comprende y analizo cada una de las etapas, para así aplicarlas en cada ejercicio donde se dio solución a cada problema propuesto

### **Zurita Cámara Juan Pablo**

Cumplimos de manera satisfactoria con el objetivo y actividades de esta práctica. Para cada problema desarrollamos un algoritmo que nos lo resolviera a manera de un tipo de pseudocódigo. Para el desarrollo de cada algoritmo se llevó a cabo el ciclo de vida de un software, sobre todo se realizó la parte de mantenimiento y evolución del algoritmo puesto a que cada integrante del equipo fue realizando correcciones.

### **Aburto López Roberto:**

Pude poner en práctica las etapas de elaboración de un software con cada uno de los ejercicios propuestos, y vi que es importante seguir cada paso para tener un programa correcto y que se nos sea más fácil su elaboración, por lo que los objetivos de la práctica se cumplieron de manera satisfactoria.

### **Téllez Cruz Brayan:**

Pudimos poner en práctica los análisis que hemos realizado y más aún el pensamiento que hemos estado desarrollando y continuaremos desarrollando a lo largo de esta asignatura y las siguientes que se relacionen. El analizar múltiples problemas nos ayuda a pensar en diferentes soluciones y a ver las cosas desde puntos alejados.