

	<b>Solución de problemas y Algoritmos</b>	
Facultad de Ingeniería	Laboratorio de docencia	

Laboratorios de computación  
salas A y B

*Profesor:* Alejandro Pimentel

*Asignatura:* Fundamentos de Programación

*Grupo:* 135

*No de Práctica(s):* 3

*Integrante(s):* Lorena Basurto Amezcu

*No. de Equipo de  
cómputo empleado:* Luxemburgo10

*No. de Lista o Brigada:* 2858

*Semestre:* 2020-1

*Fecha de entrega:* Septiembre 2, 2019.

*Observaciones:* Muy bien

CALIFICACIÓN: **10**

## 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.

## Desarrollo:

Algoritmo: un algoritmo es un conjunto ordenado e inequívoco de pasos ejecutables que definen un proceso finito.

### Ciclo de vida del software



### • **Actividad**

*Explicar las precondiciones y el conjunto de salidas de los algoritmos para:*

- *Pescar*
  - *Lavarse las manos*
  - *Cambiar una llanta*
  - *Convertir un número binario a decimal*
- 
- *Pescar*
    - ✓ Precondiciones:
      1. Contar con conocimientos previos teóricos y de preferencia prácticos sobre la pesca, o en caso de tener nulos conocimientos, contar con la compañía y asesoramiento de un experto.
      2. Contar con el tiempo y la voluntad para realizar la actividad.
      3. Tener el equipo necesario, incluyendo una caña de pescar, carnada, un bote (personal o rentado) si la actividad no se realizará a la orilla del mar o lago, botas y sombrero para pescadores, y repuestos.
      4. Encontrarse en condiciones físicas óptimas que te permitan sentarte varias

horas bajo el sol, sostener una caña de pescar durante un tiempo prolongado, así como jalar la caña de pescar con fuerza cuando el pez muerda el anzuelo.

5. Elegir adecuadamente la zona donde habrá de realizarse la actividad, y que en ésta haya peces.
6. Elegir un día con clima adecuado para pescar.

✓ Salidas:

1. Pasar un buen rato.
2. Aprender o mejorar habilidades de pesca.
3. Pescar uno o más peces.

➤ *Lavarse las manos*

✓ Precondiciones:

1. Tener el deseo o la voluntad de lavarte las manos.
2. Contar con el espacio físico, es decir, un baño, cerca.
3. Que el baño cuente con un lavabo, tubería, agua y jabón.

✓ Salidas:

1. Manos limpias y desinfectadas.
2. Higiene personal.

➤ *Cambiar una llanta*

✓ Precondiciones:

1. Tener un automóvil.
2. Tener una llanta de refacción.
3. Contar con la herramienta necesaria y adecuada dentro del automóvil (gato hidráulico, llave y birlos de seguridad).
4. Contar con los conocimientos teóricos y de preferencia prácticos para cambiar una llanta.
5. Contar con la fuerza física suficiente para cargar una llanta, para quitarla y también para colocarla.

✓ Salidas:

1. Cambiar una llanta ponchada por una de refacción.

➤ *Convertir un número binario a decimal*

✓ Precondiciones

1. Tener un número binario que se desea cambiar a decimal.
2. Conocer el algoritmo o procedimiento para hacer la conversión.
3. Contar con lápiz y papel.
4. Si se hará el cálculo con una computadora, que ésta tenga integrado el algoritmo para hacer la conversión.

✓ Salidas

1. Conocer la notación decimal equivalente de un número binario dado.

• **Actividad**

*Desarrollar un algoritmo para:*

- *Determinar si un número es positivo o negativo*
- *Obtener el mayor de dos números diferentes*
- *Obtener el factorial de un número*
  
- *Determinar si un número es positivo o negativo.*
  - ✓ Precondiciones:  
El número debe pertenecer al conjunto de los “número reales” y estar escrito en sistema decimal, por lo que cada dígito del número pertenecerá al conjunto de números: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
  - ✓ Algoritmo:
    1. Identificar al dígito que esté más lejano al primero, de derecha a izquierda. Éste será el último dígito de la cifra o número a analizar.
    2. Si del lado izquierdo del último dígito se encuentra un signo “+” o no hay nada, el número es “positivo”.
    3. Si del lado izquierdo del último dígito se encuentra un signo “-“, el número es “negativo”.
    4. Si el número es “0” (cero), éste “no es ni positivo, ni negativo”.
  
- *Obtener el mayor de dos números diferentes*
  - ✓ Precondiciones:  
El número debe pertenecer al conjunto de los “número reales” y estar escrito en sistema decimal, por lo que cada dígito del número pertenecerá al conjunto de números: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
  - ✓ Algoritmo:
    1. Si se tiene un número de un dígito (unidades), se respeta la siguiente jerarquía:  
 $0 < 1 < 2 < 3 < 4 < 5 < 6 < 7 < 8 < 9$  (9 es mayor que 8, 8 es mayor que 7, etc.)
    2. Si se tiene un número de dos o más dígitos, se respeta la misma jerarquía en su respectivo lugar (unidades, decenas, centenas, etc.):  
 $10 < 11 < 12 < 13 < 14 < 15 < 16 < 17 < 18 < 19, 9 < 19 < 29 < 39 < 49 < 59 < 69 < 79 < 89 < 99$
    3. Si se tiene un número negativo, éste siempre será menor que uno positivo:  
 $-40 < 3$
    4. Si se tienen dos números negativos, la jerarquía de los números positivos se invierte:  
 $-9 < -8 < -7 < -6 < -5 < -4 < -3 < -2 < -1$
    5. El cero siempre será menor a 1 y mayor a -1:  
 $-1 < 0 < 1$
  
- *Obtener el factorial de un número*
  - ✓ Precondiciones:  
El número debe pertenecer al conjunto de los “número naturales” y estar escrito en sistema decimal, por lo que cada dígito del número pertenecerá al conjunto de números: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
  - ✓ Algoritmo:
    1. Dado cualquier número positivo, su “factorial” será el resultado de multiplicar ese mismo número por cada uno de sus antecesores, hasta llegar al uno:  
 $5! = \text{cinco factorial} = 5 \times 4 \times 3 \times 2 \times 1 = 120$

- **Actividad**

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

- 54, -9, -14, 8, 0
- (4, 5), (-9, 16), (127, 8+4i), (7, m)
- 5, 9, 0, -3
  
- 54, -9, -14, 8, 0  
54 = “positivo”  
-9 = “negativo”  
-14 = “negativo”  
8 = “positivo”  
0 = “ni positivo, ni negativo”
  
- (4, 5), (-9, 16), (127, 8+4i), (7, m)  
(4, 5) =  $4 < 5$   
(-9, 16) =  $-9 < 16$   
(127, 8+4i) = no cumple las precondiciones => error  
(7, m) = no cumple las precondiciones => error
  
- 5, 9, 0, -3  
 $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$   
 $9! = 9 \times 8 \times 7 \times 6 \times 5 \times 4 \times 3 \times 2 \times 1 = 362\ 880$   
0! = no cumple las precondiciones => error  
-3! = no cumple las precondiciones => error

- **Actividad**

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

- Cambiar el signo de un número binario
- Hacer una suma larga binaria
  
- Cambiar el signo de un número binario
  - ✓ Precondiciones:  
Cada dígito debe pertenecer al conjunto: {0, 1}  
Aritmética básica:  $0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=10$ ,  $1+1+1=11$ ,  $1+1+1+1=10+10=100$
  - ✓ Algoritmo:
    1. Tomar un número binario de la memoria y transferirlo al registro “Alfa”.
    2. Identificar si el primer dígito (de izquierda a derecha) es un 0 o un 1. En caso de ser 0 identificarlo como “número positivo” y en caso de ser un 1 como “número negativo”.
    3. Ya sea se tenga un “número positivo” o “número negativo”, se analizará y copiará el número de derecha a izquierda de la siguiente manera: todos los números 0 y hasta llegar al primer número 1; a partir del primer número 1, los subsecuentes serán reemplazados 0 por 1 y viceversa.
    4. El nuevo número se tomará del registro “Alfa” y se guardará en el registro “Beta”.

➤ *Hacer una suma larga binaria*

✓ *Precondiciones:*

Cada dígito debe pertenecer al conjunto: {0, 1}

Aritmética básica:  $0+0=0$ ,  $0+1=1$ ,  $1+0=1$ ,  $1+1=10$ ,  $1+1+1=11$ ,  $1+1+1+1=10+10=100$

✓ *Algoritmo:*

1. Tomar el primer número binario de la memoria y transferirlo al registro "PrimerNumero".
2. Tomar el segundo número binario de la memoria y transferirlo al registro "SegundoNumero".
3. Sumar ambos números con una suma vertical, dígito por dígito.
4. Poner el número resultante en el registro "SumaDeDosNumerosBinarios".

### **Conclusiones:**

Al desarrollar un algoritmo, es muy importante tomar en cuenta las restricciones o precondiciones con las que se van a trabajar, y aclararlas desde el principio. También se debe pensar en las salidas que se buscan. También, durante su desarrollo, un algoritmo debe tener un lenguaje definido que entienda la computadora o en nuestro caso una persona. Finalmente, en esta ardua tarea, debemos siempre ser críticos y tener una metodología lógica y consistente.