



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

---

ALEJANDRO ESTEBAN PIMENTEL ALARCON

*Profesor:*

---

FUNDAMENTOS DE PROGRAMACION

*Asignatura:*

---

3

*Grupo:*

---

PRACTICA No. 3

*No de Práctica(s):*

---

MORONES FLORES INGRID YOHUALLI.

*Integrante(s):*

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o  
Brigada:*

---

PRIMER SEMESTRE.

*Semestre:*

---

3/SEPTIEMBRE/2019

*Fecha de entrega:*

---

*Observaciones:* Muy bien, la única observación es recordarte  
que todo buen reporte escrito, independientemente  
~~del formato, lleva introducción y conclusiones~~  
o secciones equivalentes

---

**CALIFICACIÓN:** 10

**OBJETIVO.**

Elaborar algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas del análisis y diseño pertenecientes al ciclo de vida del software.

**ALGORITMO.**

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

**ACTIVIDAD.**

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

**PESCAR.**

Primero que nada para pescar debemos estar o ir a algún río, laguna o algo por el estilo que contenga peces, también se debe contar con una caña para pescar, tener algún tipo de carada para los pescados, tener un anzuelo, básicamente contar con todo el equipo para pescar.

Una vez tomando en cuenta todas estas precondiciones procedemos a realizar la acción de pescar de la siguiente manera: - Colocar la carnada en el anzuelo, teniendo mucho cuidado para no lastimarse. –Lanzar el anzuelo con fuerza hacia el río o lago, con el fin de alcanzar la mayor distancia posible de la orilla. Debemos tener paciencia en este segundo paso. – Si corremos con suerte y atrapamos el pescado, se iniciará la retira de la tanza y del anzuelo del agua con rapidez para que el pez no escape. –Con el pez fuera del agua, debemos de esperar a que pare de moverse para retirar el anzuelo haciéndolo nuevamente con mucho cuidado para evitar lastimarse.

En cuanto a las posibles salidas que puede tener esta acción es que no se pudiera obtener un pez. Que se pesquen más de un pez.

**LAVARSE LAS MANOS.**

Contar con una llave por donde salga agua, o estar dentro del baño, tener a la mano una pequeña toalla para que una vez terminada la acción secase con ella, y por supuesto tener sucias las manos. Después de estas precondiciones procedemos a lavarnos las manos. –Abrimos a la llave para que salga agua. – Agarramos un poco de jabón tallándonos así las manos. –Enjuagarnos las manos con el agua. – Secarnos las manos con la toalla.

Para las salidas encontramos que se pudo haber lavado las manos sin necesidad del jabón o que no se pudiera haber lavado las manos por no cumplir con los requisitos.

**CAMBIAR UNA LLANTA.**

Debemos contar con una llanta ponchada o algo por estilo para así poder cambiar una llanta, y tener el equipo necesario, como el respuesta, un gato hidráulico, una

cuña. Tomando en cuenta estas precondiciones procedemos a cambiar la llanta. -Colocamos el carro en un lugar seguro. –Sacar las herramientas requieras . – Ponemos el freno de mano y colocamos la cuña. –Aflojamos las tuercas. – Levantamos el carro con el gato hidráulico. –Quitamos las tuercas y la copa del carro. –Se retira la llanta. –Y se coloca el repuesto.

En cuanto a las posibles salidas se tiene que pudo haberse cambiado mas de una llanta y que no se haya podido llevar la acción por no cumplir con alguna precondición.

### **CONVERTIR UN NÚMERO BINARIO A DECIMAL.**

Primero debemos de contar con un número binario para convertirlo a decimal, conocer la base en la que se está trabajando, saber convertir de binario a decimal, conocer la equivalencia de las potencias.

Posteriormente según el número binario que tengamos que convertir a decimal lo vamos elevando a las potencias correspondientes y lo multiplicamos por el 0 o el 1, según sea el caso, para al final sumar las potencias y llegar al número decimal. En cuanto a las posibles salidas que se podría obtener tenemos que lleguemos exitosamente al número, que no lo obtengamos por no cumplir con las precondiciones.

### **2da ACTIVIDAD.**

Desarrollar los algoritmos para:

#### **DETERMINAR SI UN NUMERO ES NEGATIVO O POSITIVO.**

Para este caso vamos a pensar lo siguiente:

Tenemos que tener datos de entrada que son los siguientes:

- Tenemos un número real “x”. – Comparamos el número “x” con el número cero, si el número “x” es mayor o igual a cero esto quiere decir que es positivo, en el caso contrario si el número “x” es menor que cero, decimos que este número será negativo.

#### **OBTENER EL MAYOR DE DOS NUMEROS DIFERENTES.**

Para este caso planteamos lo siguiente.

Por supuesto datos de entrada.

- Tenemos dos números “x” y “y” que son reales. – Comparamos estos números  $x > y$ . –Para este caso tenemos dos posibles respuestas las cuales podrían ser que “x es mayor que y”. –Que la respuesta fuera no, simplemente llega al final.- O que  $x = y$  en este caso ponemos esta incógnita,

y las dos posibles respuestas serían “x es igual a y” o que “y es mayor que x”.

### **OBTENER EL FACTORIAL DE UN NÚMERO.**

Necesitamos los datos de entrada

Tenemos al número “x” positivo.

-Si “x” es igual a cero, su factorial es 1. – Multiplicamos el número “x” desde el número 1. –Si “x” es igual a 1, su factorial será 1. –Para sacar la factorial de los números debemos multiplicar siguiendo la siguiente regla:  $a! = 1 * (1+1) * (1+2) \dots * (x-1) * (x)$ . Esto quiere decir que desde uno multiplicaremos el número por el siguiente número, por el número siguiente hasta llegar a multiplicar por el valor de “x”. – Posteriormente obtendremos el factorial de “x” al hacer la multiplicación correspondiente.

### **VERIFICAR SUS ALGORITMOS ANTERIORE, “EJECUTARLOS” PASO A PASO CON LOS SIGUIENTES VALORES:**

- 54, -9, -14, 8, 0

Para este caso determinar si es positivo o negativo el algoritmo aplica para dichos números ya que son reales y como dice el algoritmo procedemos a hacer la comparación para cada uno de ellos.

- (4,5), (-9,16), (127, 8 +4i), (7,m)

En este caso solo aplica para los los números que están en pareja, pues en los otros casos no se podría realizar ya que el algoritmo no contiene variables con valores de numero asignados para “i” y “m” y **no se podría llevar a cabo lo** solicitado.

Lo importante aquí es observar que las precondiciones "atrapan" estos casos que se salen de la norma.

- 5, 9, 0, -3

En los casos en donde sin problema alguno se podría llevar el algoritmos es en el 5, 9 ya que cumplen con todos las precondiciones, pero en el caso del -3 no aplicaría ya que el algoritmos solo se utiliza para los positivos. En el caso del 0 el algoritmo aplica por la regla que pusimos que es la siguiente Si “x” es igual a cero, su factorial será 1.

### **3ra ACTIVIDAD.**

Desarrollar algoritmos propios de un procesador (asignando registros genéricos)

para :

### **CAMBIA EL SIGNO DE UN NÚMERO BINARIO.**

Tenemos que:

1. Disco duro: Manda registros "x" y "z" a la RAM que solicito Control.
2. Control: De la RAM extrae los registros negativos "x" y "z" y los manda a los registros del procesador vacíos en lista de espera.
3. Control: Manda "x" y "z" a ALU para cambiar el signo de bit negativo 1 a el signo de bit positivo 0.
4. ALU: Recibe "x" y "z" y aplica el comando NOT después del primer número uno del lado derecho yendo en el orden de derecha a izquierda.
5. ALU: Manda registros "x" y "z" a Control.
6. Control: Recibe "x" y "z" y los incorpora a los lugares principales del procesador en espera para mandar a la RAM.
7. RAM: Recibe los registros "x" y "z" y los incorpora en el disco duro.
8. Disco duro: Recibe "x" y "z" de la RAM y los almacena.

### **HACER UNA SUMA LARGA BINARIA**

1. Disco duro: Manda registros "a", "b", "c", "d", "e" y "f" a la RAM que solicito Control.
2. Control: De la RAM extrae los registros "a", "b", "c", "d", "e" y "f" y los manda a los registros del procesador vacíos en lista de espera.
3. Control: Manda "a", "b", "c", "d", "e" y "f" a ALU para cambiar los números negativos a positivos si es que fuera el caso, si no saltar al paso 6.
4. ALU: Recibe de "a", "b", "c", "d", "e" y "f" solo los números que se encuentren negativos, es decir con el bit de signo 1 y aplica el comando NOT a cada registro después del primer número uno del lado derecho yendo en el orden de derecha a izquierda.
5. ALU: Manda los registros ya cambiados de signo a Control.
6. Control: Manda "a", "b", "c", "d", "e" y "f" a ALU para hacer una suma binaria.
7. ALU: Recibe "a", "b", "c", "d", "e" y "f" y aplica el comando XOR a los registros de dos en dos hasta llegar a un solo resultado en el orden de derecha a izquierda.
8. ALU: Manda solo un registro llamado "z", que es la suma de los registros "a", "b", "c", "d", "e" y "f" a Control.
9. Control: Recibe "z" y lo incorpora a un lugar principal del procesador en lista de espera para mandar a la RAM.
10. RAM: Recibe el registro "z" y lo incorpora en el disco duro.
11. Disco duro: Recibe "z" de la RAM y lo almacena.

Para concluir podemos decir que se llegó al objetivo, comprendiendo que los algoritmos son sumamente necesarios para realizar todo tipo de procesos que deseemos o que requiramos hacer en una computadora.