



# Programación lógica Y funcional

---

## Ejercicios

**Alumno:** Jesús Armando Juárez Fernández.

**Carrera:** Ing. En sistemas computacionales.

**Nombre de la asignatura:** Programación lógica y funcional.

**Unidad:** 1-. Conceptos Fundamentales.

**Semestre:** 6

**Grupo:** 603 A.

**Fecha:** 08/03/2022

## Índice

### Contenido

Introducción .....	1
Desarrollo de la práctica.....	2
Resultados.....	3
Conclusiones .....	6
Referencias bibliográficas .....	7

## Índice de imágenes

Figura 1. Solución del ejercicio 1. ....	3
Figura 2. Solución del ejercicio 2. ....	3
Figura 3. Solución del ejercicio 3. ....	3
Figura 4. Solución del ejercicio 4. ....	4
Figura 5. Solución del ejercicio 5. ....	4
Figura 6. Solución del ejercicio 6. ....	4
Figura 7. Solución del ejercicio 7. ....	5
Figura 8. Solución del ejercicio 8. ....	5

## Introducción

La primera unidad correspondiente a la asignatura de programación lógica y funcional consiste en conocer y comprender los conceptos fundamentales existentes en dicha asignatura que, de hecho, es de suma importancia tenerlos bien entendidos, puesto a que serán relevantes en las posteriores unidades de la asignatura donde serán manejados.

En si la primera una unidad es la introducción general de la asignatura, es la base de toda la asignatura que será útil para poder partir, dicha base ira evolucionando con el paso de las unidades, al conocer más conceptos, ideas, herramientas, conocimientos, entre otros.

Partiendo por conocer los distintos estilos de programación existentes, si no es que los más usados actualmente, cabe mencionar que surgen cuatros estilos de programación diferentes, sin contar los derivados de algunos de los estilos, entre ellos se destacan: la programación imperativa (donde surge también la programación estructurada, la programación procedimental y la programación modular), la programación declarativa, la programación orientada a objetos y la programación reactiva.

Una vez entendido cuales son los tipos de programación habituales y en qué consisten cada uno de ellos, es necesario también tener en cuenta la importancia de determinados parámetros que también son muy fundamentales en la programación lógica y funcional, donde aquí entra la evaluación de expresiones, donde es necesario conocer cómo funciona, en que orden trabaja, como puede verse alterado su orden con ciertos operandos, entre otros aspectos más.

Un parámetro que también es esencial tomar a consideración, no solamente por conocer si no también para el momento en el que se está codificando, es conocer los tipos de datos que se están manejando, saber cuál es la función que ejerce para determinados contextos, cuáles son sus rangos y sobre todo, saber cuánto espacio de almacenamiento tomara en el sistema de cómputo, debido a que existen ciertos casos que al usar un mismo tipo de dato sin conocer el peso de este, al momento de llevar a cabo su ejecución, el espacio se disparará potencialmente causando que la ejecución muestre anomalías, por ello es importante saber el tipo de dato que se manejará.

Un elemento que también será muy presente en la programación lógica y funcional es la disciplina de tipos, desde una perspectiva general, la disciplina de tipos se encarga de la comprobación de la posibilidad de errores durante el tiempo de compilación, lo anterior es de forma estática, en otras de palabras, si un lenguaje de programación es capaz de detectar los errores de tipos, es porque cuenta con la disciplina de tipos.

El último concepto por considerar dentro de la primera unidad de dicha asignatura, son las funciones, en resumidas palabras, se trata de un fragmento de codificación correspondiente a una codificación principal.

Lo anterior sería todo respecto a los conceptos contenidos en la primera unidad perteneciente a la asignatura de programación lógica y funcional, la siguiente evidencia a entregar consiste en una serie de ejercicios en conjunto con un mapa conceptual, ambos son necesarios para la acreditación de la unidad antes mencionada.

Dichos ejercicios consisten en diseñar funciones que atiendan determinadas necesidades, la particularidad de dichas funciones es de diseñarlas de la forma más optimizada posible, de alguna forma llevarlos a cabo de una manera general.

Es por ello por lo que, en determinado punto del semestre se comenzaron a desarrollar una serie de funciones con tal de resolver cada uno de los ejercicios asignados a lo largo del transcurso, una vez estén completamente desarrollados, es necesario mostrar el proceso que se llevó a cabo para su solución y una breve descripción.

Es por ello por lo que primeramente se mostrarán cada uno de los ejercicios planteados a lo largo de la primera unidad, esto con tal de entender que es lo que se desarrollará y tener una idea de lo que se va a hacer, una vez presentados los ejercicios asignados, se procederá a mostrar las funciones diseñados para cada uno.

En resumidas palabras, esta unidad es la base para entender de que tratará la asignatura en conjunto de ejercicios para agarra más hilo sobre lo que tratará la asignatura en cuestión.

## Desarrollo de la práctica

A continuación, se presentan los siguientes ejercicios planteados a lo largo de la primera unidad de la asignatura:

- Diseñar una función que permita calcular el porcentaje de aprobación de un grupo de  $n$  alumnos.
- Diseñar una función para validar la entrada de un numero solo se aceptan números positivos.
- Diseñar una función para calcular e imprimir una tabla de multiplicar definida por el usuario.
- Diseñar una función para calcular la ley de Ohm.
- Diseñar una función para calcular todas las identidades trigonométricas.
- Diseñar una función para calcular una multiplicación a través de sumas sucesivas.
- Diseñar una función que genere una serie de Fibonacci hasta el valor máximo indicado por el usuario.
- Diseñar una función para ubicar el mayor dato numérico almacenado en un arreglo de 10 posiciones.

Tal como se aprecia, cada uno de los presentes ejercicios que se muestran, se les debe diseñar una función, tiene que ser lo más optimizada posible, a continuación, en el apartado de resultados, se muestran cuáles fueron los resultados que surgieron en base al desarrollo.

Por último, cabe agregar que la sintaxis que se utilizó para el diseño de las funciones fue la siguiente:

```
nombreFuncion([listaDeArgumentosDeEntrada]){  
  código  
  ....  
  ....  
  ....  
  retorno variable  
}
```

## Resultados

1.- Diseñar una función que permita calcular el porcentaje de aprobación de un grupo de n alumnos.

```
/*Diseñar una funcion que permita calcular el porcentaje
de aprobacion de un grupo de n alumnos*/

calcuporcen(n, aprobados, porcentaje){
porcentaje=aprobados*100/n
retorno porcentaje
}
```

Figura 1. Solución del ejercicio 1.

2.- Diseñar una función para validar la entrada de un numero solo se aceptan números positivos.

```
/*Diseñar una funcion para validar la entrada de
un numero solo se aceptan numeros positivos*/

NumPos(num){
mientras(num<1)
validarNumero(dato*-1)
retornar dato
}
```

Figura 2. Solución del ejercicio 2.

3.- Diseñar una función para calcular e imprimir una tabla de multiplicar definida por el usuario.

```
/*Diseñar una función para calcular e imprimir
una tabla de multiplicar definida por el usuario*/

Tabla(i, tabla){
Para i=1 hasta 10 con paso 1 hacer
retornar tabla*i
}
```

Figura 3. Solución del ejercicio 3.

4.- Diseñar una función para calcular la ley de Ohm.

```
/*Diseñar una funcion para
calcular la ley de Ohm*/

LeyOhm(dato1, dato2){
si ley=dato1/dato2
si no
ley=dato1*dato2
retorno ley
}
```

Figura 4. Solución del ejercicio 4.

5.- Diseñar una función para calcular todas las identidades trigonométricas.

```
/*DISEÑAR UNA FUNCION PARA CALCULAR
TODAS LAS IDENTIDADES TRIGONOMETRICAS*/

Trigo(dato1, dato2){
retorna dato1/dato2
}
```

Figura 5. Solución del ejercicio 5.

6.- Diseñar una función para calcular una multiplicación a través de sumas sucesivas.

```
/*DISEÑAR UNA FUNCION PARA CALCULAR UNA
MULTIPLICACION A TRAVES DE SUMAS SUCEASIVAS*/

sumasuc(num1, num2, suma, i){
    suma=0
    Para i desde num2
        suma = suma+num1
    retorno suma
}
```

Figura 6. Solución del ejercicio 6.

7.- Diseñar una función que genere una serie de Fibonacci hasta el valor máximo indicado por el usuario.

```
/*DISEÑAR UNA FUNCIÓN QUE GENERE UNA SERIE DE FIBONACCI  
HASTA EL VALOR MAXIMO INDICADO POR EL USUARIO*/  
  
fibo(valor){  
    dato1=0  
    dato2=1  
    para i=1 hasta valor con paso 1 hacer  
        serie=dato1+dato2  
        dato1=dato2  
        dato2=serie  
    retornar serie  
}
```

*Figura 7. Solución del ejercicio 7.*

8.- Diseñar una función para ubicar el mayor dato numérico almacenado en un arreglo de 10 posiciones.

```
/*DISEÑAR UNA FUNCIÓN PARA UBICAR EL MAYOR DATO  
NUMÉRICO ALMACENADO EN UN ARREGLO DE 10 POSICIONES*/  
  
mayor(numero[]){  
    mayordato=0  
    para i=1 hasta 10 con paso 1 hacer  
        si (numero[i]>mayordato)  
            mayordato=numero[i]  
    retornar mayordato  
}
```

*Figura 8. Solución del ejercicio 8.*



## Conclusiones

De esta forma concluye el documento que incluye los ejercicios propuestos, así como su respectiva solución, dichos ejercicios fueron planteados a lo largo de la primera unidad correspondiente a la asignatura de programación lógica y funcional.

La resolución de estos ejercicios tienen el objetivo de ser una introducción de lo que es la asignatura ya antes mencionada, es de suma importancia entender la forma de solucionar estos ejercicios ya que estos son la idea base que sirven para partir hacia las posteriores unidades, entendido lo que se hizo en los ejercicios será más fácil de entender que es lo que se hará próximamente, con el fin de tener una idea de lo que se hará y no presentar algún tipo de inconveniente al llegar a dichas unidades.

Por otro lado, e independientemente de tomar una base de lo que resta del semestre, la resolución de estos ejercicios también tiene la finalidad de impulsar la deducción, saber analizar cada problema que se presenta, es necesario saber identificar que se necesita para resolver un determinado problema, identificar las ideas de donde se pueden partir y obtener la capacidad de optimizar las soluciones de la mayor manera posible, en general su objetivo es obtener la capacidad de analizar correctamente.

De esta forma, en base a la entrega de los ejercicios propuestos y el mapa conceptual, han quedado completados los criterios de entrega de evaluación de la primera unidad, más precisamente en el criterio de conforman a las tareas.

También cabe añadir que, en base a los ejercicios planteados en la mencionada unidad, ha quedado clara la forma de desarrollar soluciones ante determinados ejercicios y optimizarlos de la mayor forma posible, la base principal para poder encontrar su solución es verlo desde una perspectiva general, en otras palabras, generalizar lo más posible las ideas por las que se rige el ejercicio de esta forma procediendo con su respectiva solución.

No solamente ha quedado completamente claro en base a los ejercicios analizados, si no también respecto a cada uno de los contenidos integrados en la unidad, que también es esencial que haya quedado comprendido, al igual que los ejercicios, los conceptos analizados también son fundamentales de entender con tal de reconocer que es lo que se está o se estará manejando en determinado punto.

Como lo ha sido, por ejemplo, los diferentes estilos de programación que habitualmente se utilizan, en que consisten cada uno de ellos y como es que en determinados estilos todavía surgen más estilos los cuales de igual forma tienen una determinada función.

También una determinada serie de parámetros que como antes se mencionó, son importantes de reconocer ya que por ello se rige completamente la programación lógica y funcional.

Entre ellos, esta el evaluador de expresiones, que se rige por una serie de factores para poder llevar a cabo un análisis, como lo es el orden en que las expresiones están ordenadas y los operandos que estén colocados, cabe recordar que el cambio o eliminación de un operando puede cambiar drásticamente el orden.

Es esencia también reconocer que es la disciplina de tipos, que como con anterioridad se mencionó, si un lenguaje de programación es capaz de detectar los errores de tipos, es porque cuenta con disciplina de tipos.

Por último, pero no menos importante, identificar que son las funciones y que es lo que hacen, en pocas palabras, son una sección de código perteneciente a un código principal, ese fragmento de código se encarga de efectuar una operación en particular, dependiendo de lo que se esté programando.

Lo anterior mencionado, son todas las ideas base que engloban por completo a la unidad de dicha asignatura.

Desde un punto de vista general, la unidad uno ha servido sólidamente para entender o por lo menos llevar una idea de los temas que se analizarán y se desarrollarán próximamente en las unidades posteriores en lo que resta del semestre, por lo que la entrega de este documento concluye de forma concreta la primera unidad.

## Referencias bibliográficas

- Valle Pérez, R. (2017, 16 febrero). Unidad 1 Conceptos Fundamentales Programación Lógica y Funcional. Wordpress. <https://programacion231.wordpress.com/2017/02/16/unidad-1-conceptos-fundamentales-programacion-logica-y-funcional/>
- Colorado López, T. (2014, 13 febrero). EVALUACIÓN DE EXPRESIONES. Wordpress. <https://programacionlogica-y-funcional.wordpress.com/2014/02/12/evaluacion-de-expresiones/>
- Moreno, L. C. (2019, 21 febrero). 1.2.2 TIPOS DE DATOS. Programación Lógica y funcional. <https://lucydelcarmenleonmoreno.blogspot.com/2019/02/123-tipos-de-datos.html>
- Pérez Y Bernardo, M. Á. (2016, 6 marzo). 1.4 Disciplina tipos. Programación lógica y funcional. <http://programacion-logica-funcional.blogspot.com/2016/03/14-disciplina-de-tipos.html>
- López Figueroa, A. B. (2019, 21 febrero). 1.2.4 Funciones. Programación lógica y funcional. <https://alexisbladimirlopezfigueroa.blogspot.com/2019/02/124-funciones.html>