



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* M.I. Marco Antonio Martínez Quintana

*Asignatura:* Fundamentos de Programación

*Grupo:* 3

*No de Práctica(s):* 05

*Integrante(s):* Ramirez Garcia Diego Andres

*No. de Equipo de  
cómputo empleado:* No Aplica

*No. de Lista o Brigada:* 35

*Semestre:* Primero

*Fecha de entrega:* 02/Noviembre/2020

*Observaciones:* Pertinente para el correcto desarrollo del código de nuestros futuros

proyectos puesto facilita la codificación y el pensamiento abstracto.

**CALIFICACIÓN:** \_\_\_\_\_

## Objetivos

Elaborar pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

## Introducción

El pseudocódigo es una representación algorítmica elemental para que la solución de un problema (algoritmo) pueda ser codificada. Este muestra en forma de texto los pasos a seguir para solucionar un problema por lo que posee una sintaxis propia al igual que los diagramas de flujo.

### Sintaxis de pseudocódigo

A continuación, se describen las reglas semánticas y sintácticas más importantes:

1. Alcance del programa: Todo pseudocódigo está limitado por las etiquetas de **INICIO y FIN**. Dentro de estas etiquetas se deben escribir todas las instrucciones del programa.
2. Palabras reservadas con **mayúsculas**: Todas las palabras propias del pseudocódigo deben de ser escritas en mayúsculas.
3. Sangría o **tabulación**: El pseudocódigo debe tener diversas alineaciones para que el código sea más fácil de entender y depurar.
4. Lectura / escritura: Para indicar lectura de datos se utiliza la etiqueta **LEER**. Para indicar escritura de datos se utiliza la etiqueta **ESCRIBIR**. La lectura de datos se realiza, por defecto, desde el teclado, que es la entrada estándar del sistema. La escritura de datos se realiza, por defecto, en la pantalla, que es la salida estándar del sistema.

## Variables

Existen diferentes tipos de datos posibles a asignar a una variable:

- ENTERO -> valor entero positivo y/o negativo
- REAL -> valor con punto flotante y signo
- BOOLEANO -> valor de dos estados: verdadero o falso
- CARACTER -> valor tipo carácter
- CADENA -> cadena de caracteres

La declaración de variables la definen un nombre, seguido de dos puntos, seguido del tipo de dato, es decir:

**contador: ENTERO**

Para declarar más de una variable del mismo tipo empleamos arreglos, indicando la cantidad:

**contador [5]: ENTERO // 5 variables de tipo entero**

Para contener uno o más tipos de datos simples diferentes utilizamos un registro o estructura:

```
domicilio:REG
  calle: CADENA
  número: ENTERO
  ciudad: CADENA
FIN REG
```

Por último tenemos las variables constantes a partir de la asignación de CONST, es decir, un tipo de variable al que se le asigna un valor que no queremos que cambie.

**NUM\_MAX := 1000: REAL, CONST**

## Notación de camello

Se emplea para nombrar variables y nombres de funciones. Existen dos tipos de notaciones de camello: lower camel case y upper camel case.

- a) lower camel case: la primera letra de la variable inicia con minúscula
- b) upper camel case: la primera letra de la variable inicia con mayúsculas

```
// variables
realAreaDelTriangulo: REAL // lower camel case
EnteroRadioCirculo: REAL // upper camel case

// funciones
calcularArea() // lower camel case
ObtenerPerimetro()// upper camel case
```

## Estructuras de control de flujo

Permiten la ejecución condicional y la repetición de un conjunto de instrucciones; existen tres estructuras de control: secuencial, condicional y repetitivas o iterativas:

**Secuencial:** se realizan en el orden en el que están escritas.

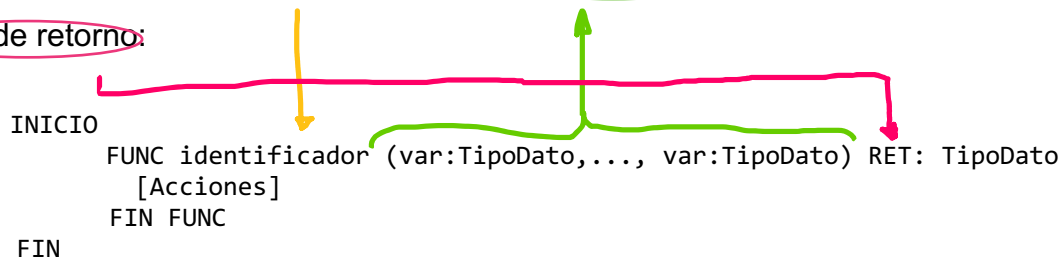
**Condicional:** SI, SI-DE LO CONTRARIO, SELECCIONAR-CASO.

**Iterativas o repetitivas:** MIENTRAS y HACER- MIENTRAS.

## Funciones

Cuando la solución de un problema es muy compleja se suele dividir un problema en varios subprocesos más sencillos que juntos forman la solución completa.

Está constituida por un nombre, de cero a n parámetros de entrada (variables) y un valor de retorno:



## Desarrollo de Actividades

Realizar el pseudocodigo que determine el color del semáforo COVID en base a una muestra de 100 individuos:

- Si hay más de 80 individuos con COVID el color del semáforo es rojo
- Si hay de 51 a 80 individuos con COVID el color del semáforo es naranja
- Si hay de 1 a 50 individuos con COVID el color del semáforo es amarillo
- Si no hay individuos con COVID el color del semáforo es verde.

INICIO

```
base:ENTERO //muestra
cons:ENTERO //contagiados
base:=100
```

```
ESCRIBIR "Número de contagiados"
LEER cons //contagiados
```

```
sem:= base-cons //sem significa semaforo
```

```
SELECCIONAR (sem) EN
    CASO [0 - 20) ->
        ESCRIBIR "Semaforo Rojo."
    CASO [20 - 49] ->
        ESCRIBIR "Semaforo Naranja."
    CASO [50 - 99] ->
        ESCRIBIR "Semaforo Amarillo."
    CASO 100 ->
        ESCRIBIR "Semaforo Amarillo."
    DEFECTO ->
        ESCRIBIR "Opción Invalida."
FIN SELECCIONAR
```

FIN

## Desarrollo de Actividades

Realizar un diagrama de flujo que calcule dado un número el cálculo de su factorial:

Ejemplo:

1.  $1! = 1$
2.  $2! = 2$
3.  $3! = 6$
4.  $4! = 24$

INICIO

```
num:ENTERO //número
fact:ENTERO //factorial
fact:=1
```

```
    ESCRIBIR "Factorial de un número"
    LEER num //número
```

```
MIENTRAS num > 1 ENTONCES
    fact = fact*num
    nun := num-1
FIN MIENTRAS
```

```
    ESCRIBIR "fact"
```

FIN

## **Conclusiones**

El desarrollo de un proyecto debe seguir una metodología para conseguir resultados estructurados y evitar errores en el producto final. Por lo anterior el pseudocódigo representa parte fundamental del desarrollo adecuado de un software, ya que nos permite estructurar de mejor manera la codificación; al ser la representación gráfica del algoritmo al mismo tiempo nos podemos percatar de los errores dentro de él y otras maneras más eficientes de llegar al mismo resultado.

## Referencias

*“Ingeniería en Computación(2018)Guía de estudio práctica  
04:Diagramas de Flujo, Manual de Prácticas del Laboratorio de  
Fundamentos de Programación”*