

# Fundamentos de Programación

Conceptos Básicos

Oscar Hernando Arenas Arenas  
oharenas@udem.edu.co

# Contenido

- Computador
  - Hardware
  - Software
- Algoritmo
- Seudocódigo
- Palabra reservada
- Programa
- Lenguaje de Programación
- Fases en la solución de problemas
- Estructura de un algoritmo

# Computador



- Dispositivo electrónico utilizado para procesar datos y obtener resultados.
- Los datos y la información se pueden introducir en la computadora mediante dispositivos de entrada (teclado, mouse, etc.) y a continuación se procesan para producir una salida o resultado.
- Está conformado por componentes físicos (hardware) y componentes lógicos (software).



# Organización física de un computador

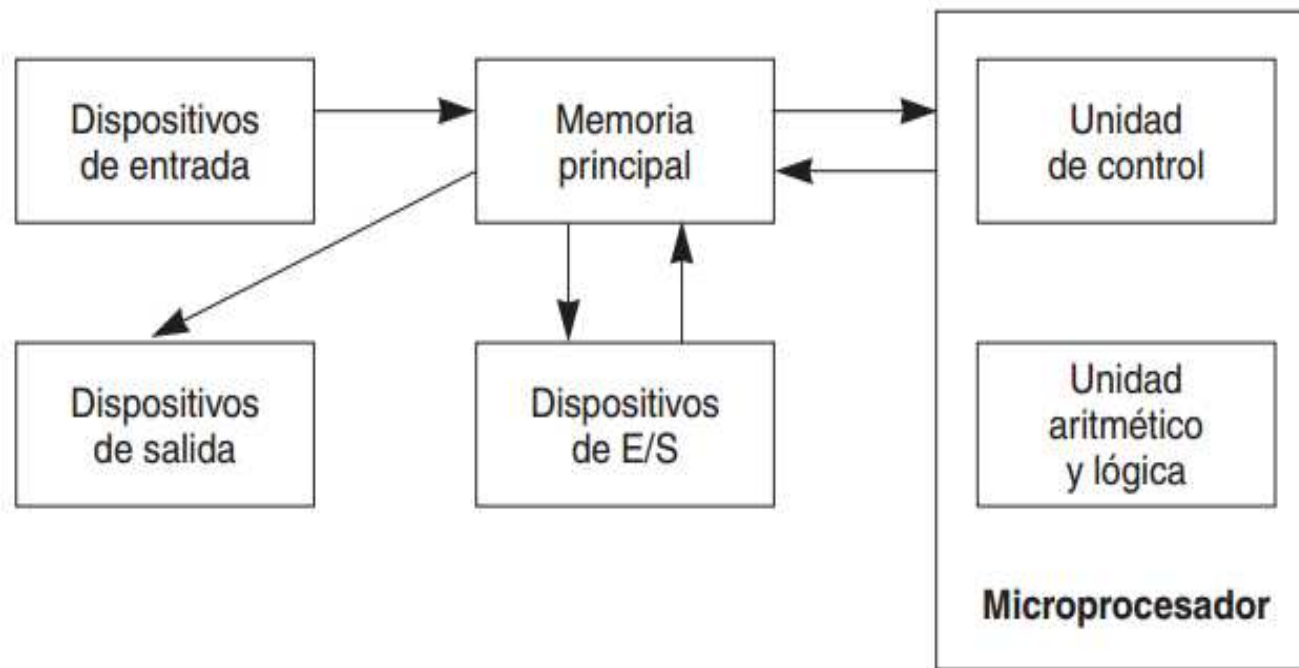


Imagen tomada de (Joyanes, 2008)

# Hardware (1/2)

Componentes físicos que constituyen la computadora, junto con los dispositivos que realizan las tareas de entrada y salida de datos.

- Dispositivos de entrada/salida o periféricos: Permiten la comunicación entre el computador y el usuario.
- Los dispositivos de entrada sirven para introducir datos (información) en el computador para su proceso. Los datos se leen de los dispositivos de entrada y se almacenan en la memoria principal. Dispositivos de entrada comunes: teclado, mouse, touchpad, lápices ópticos, joystick, gamepad, lectores de códigos de barras, escáneres, micrófonos, etc.
- Los dispositivos de salida permiten representar los resultados (salida) del proceso de los datos. El dispositivo de salida típico es la pantalla o monitor. Otros dispositivos de salida son: impresoras (imprimen resultados en papel), trazadores gráficos (plotters), reconocedores (sintetizadores) de voz, altavoces, etc.
- Dispositivos de entrada/salida y dispositivos de almacenamiento masivo o auxiliar (memoria externa): unidad de discos (disquetes, CD-ROM, DVD, cintas, discos duros externos, etc.), videocámaras, memorias flash, USB, pantalla táctil, Google Glass, etc.

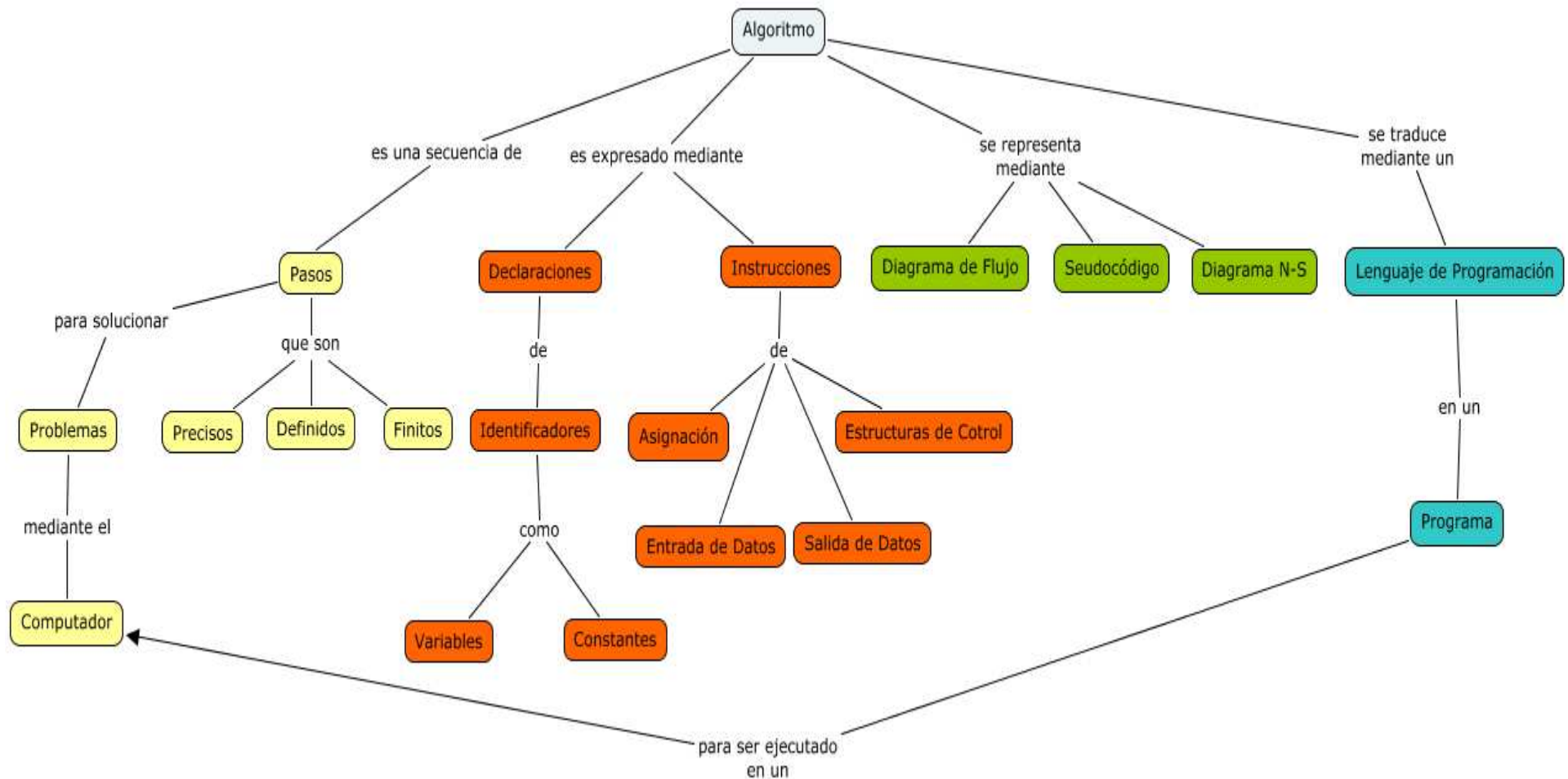
# Hardware (2/2)

- Memoria principal: La memoria central o simplemente memoria (interna) se utiliza para almacenar información. En general, la información almacenada en memoria puede ser de dos tipos: instrucciones de un programa y datos con los que operan las instrucciones.
- Procesador: Unidad Central de Proceso (Central Processing Unit, **CPU**, en inglés). También conocido como microprocesador, dirige y controla el proceso de la información realizado por el computador. Sus componentes principales son:
  - Unidad de Control: Coordina las actividades de la computadora y determina qué operaciones se deben realizar y en qué orden.
  - Unidad de Aritmética y Lógica: Realiza operaciones aritméticas y lógicas, tales como suma, resta, multiplicación, división y comparaciones.

# Software

- Conjunto de programas que indican al computador las tareas que debe realizar.
- Los computadores procesan datos bajo el control de un conjunto de instrucciones denominados programas.
- Los programas son **algoritmos** expresados mediante algún lenguaje de programación.
- A la persona que escribe programas se le conoce como programador.

# Mapa Conceptual: Algoritmo





# Algoritmo

- Método para solucionar problemas mediante una serie de pasos precisos, definidos y finitos.
- Diseñar un buen algoritmo exige un buen nivel de abstracción, lógica, imaginación, creatividad, dominio de las técnicas de programación y conocimiento del área del problema que se intenta resolver.
- Por esto, es que el diseño de algoritmos se considera tanto ciencia como arte.

# Características de los Algoritmos

- Un algoritmo debe ser *preciso*. Indicar el orden de realización de cada paso.
- Un algoritmo debe ser *definido*. Si se sigue un algoritmo dos veces, con los mismos datos de entrada, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser *finito*. Si se sigue un algoritmo, se debe terminar en algún momento, es decir, debe tener un numero finito de pasos.

# Partes de un Algoritmo

- **Entrada:** Se suministran al algoritmo los datos a partir de los cuales se desean obtener unos resultados.
- **Proceso:** Esos datos son modificados mediante las instrucciones que contiene.
- **Salida:** Debe suministrar esa información procesada al usuario o a otro programa que haga uso de ella.

# Seudocódigo

- Es un lenguaje de especificación (descripción) de algoritmos.
- Esta formado por palabras reservadas, instrucciones, identificadores y operadores.
- En principio, no puede ser ejecutado por un computador.
- Su uso hace que el paso de codificación sea fácil.

# Palabra Reservada

Es una secuencia de caracteres que tiene un significado especial y preciso dentro de un lenguaje especificación de algoritmos o un lenguaje de programación, por tanto, no puede ser utilizada como nombre de una variable, constante o subalgoritmo.

Palabras reservadas del pseudocódigo que utilizaremos:

- algoritmo, variables, entero, real, logico, caracter, cadena, infinito, verdadero, falso, inicio-fin, si-entonces-si\_no-fin\_si, mientras-hacer-fin\_mientras, para-hasta-paso-fin\_para, pi
- funcion, devolver

# Subalgoritmos

Conjunto de declaraciones e instrucciones que llevan a cabo **una** tarea específica. Se dividen en procedimientos y funciones. Utilizaremos los siguientes en el diseño de algoritmos, sin considerar como están definidos:

- **Procedimientos:**

- `muestre()`: Utilizado para indicar el envío de datos (información) a un dispositivo de salida del computador y ser visualizados por el usuario. Muestra o imprime valores o variables por pantalla.
- `lea()`: Se utiliza para indicar que el usuario debe ingresar datos a un algoritmo por medio de algún dispositivo de entrada del computador. Los datos ingresados se asignan a variables del algoritmo.

- **Funciones:**

- Las listadas en la tabla 1.

# Funciones Matemáticas

Nombre función	Seudocódigo	MATLAB
Aleatorio	aleatorio()	rand()
Aleatorio entero	aleatorio(x)	randi(x)
Coseno	cos(x)	cos(x)
Exponencial	exp(x)	exp(x)
Logaritmo natural	ln(x)	log(x)
Piso	piso(x)	floor(x)
Raíz cuadrada	raiz2(x)	sqrt(x)
Redondear	redondear(x)	round(x)
Seno	sen(x)	sin(x)
Tangente	tan(x)	tan(x)
Techo	techo(x)	ceil(x)
Valor absoluto	abs(x)	abs(x)

Tabla 1. Funciones matemáticas básicas internas.

# Programa

- Conjunto de instrucciones internas utilizadas para ejecutarse en un computador y producir un resultado.
- Los programas se hacen (escriben) con lenguajes de programación.



# Lenguaje de Programación

- Es un lenguaje diseñado para ser leído y escrito por humanos y ejecutado por computadoras.
- Algunos lenguajes de programación son: C, C++, C#, Java, Python, JavaScript, PHP, Fortran, Objective-C, Perl, Ruby, Pascal.
- Escribiremos nuestros programas con el lenguaje de programación MATLAB.

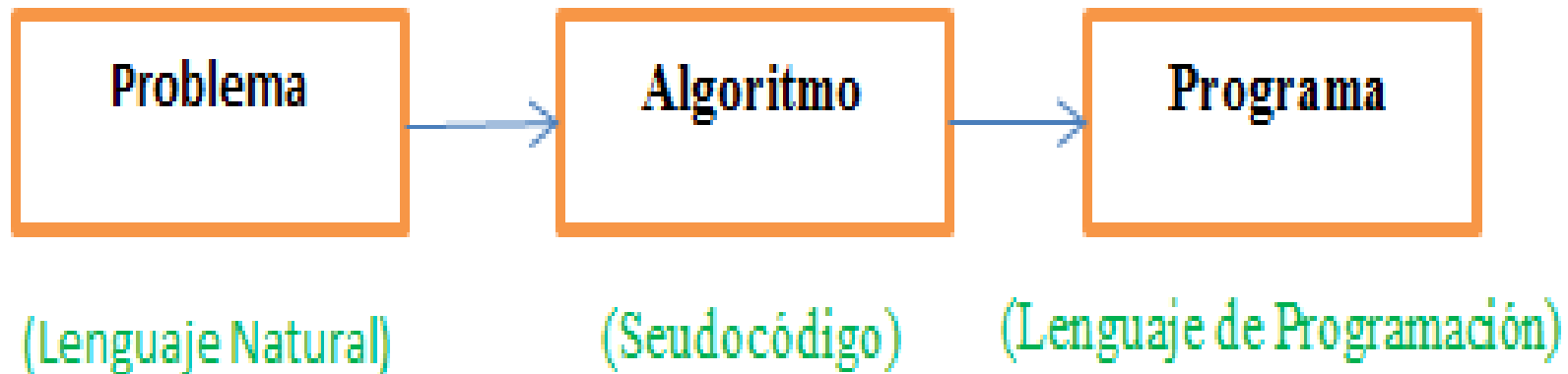
# Desventajas del lenguaje natural

- Ambigüedad. Una misma oración puede ser interpretada de varias formas por diferentes personas.
- Verbosidad. Existen muchas palabras para expresar una solución.

# Fases en la solución de problemas

1. Definición o análisis del problema
  - Entender el problema
  - Determinar datos entrada
  - Proceso
  - Determinar datos salida
2. Diseño del algoritmo
3. Transformación del algoritmo en un programa (Codificación)
4. Ejecución y validación del programa

# Herramientas para la solución de problemas



# Estructura de un Algoritmo

algoritmo <NombreDelAlgoritmo>

variables

<TipoDeDato>: Lista nombre variables

⋮

<TipoDeDato>: Lista nombre variables

inicio

<Instruccion 1>

<Instruccion 2>

⋮

<Instruccion n>

fin

# Ejemplo 1: Algoritmo para sumar dos números

algoritmo SumarDosNumeros

variables

real: a, b, c

inicio

muestre('SUMAR DOS NUMEROS')

muestre('Entre el primer numero:')

lea(a)

muestre('Entre el segundo numero:')

lea(b)

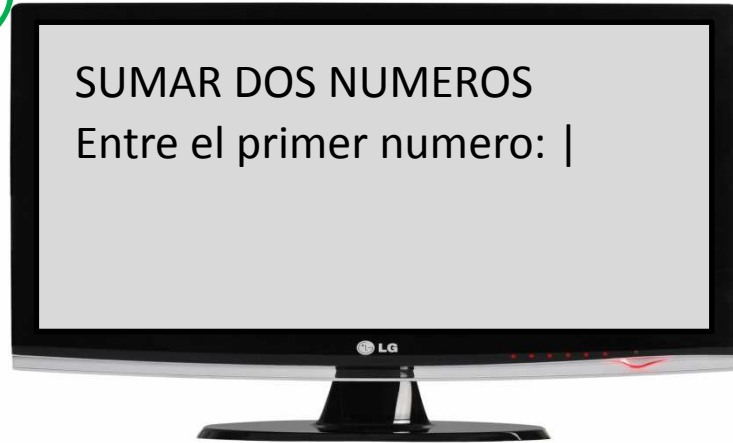
$c = a + b$

muestre('La suma es: ', c)

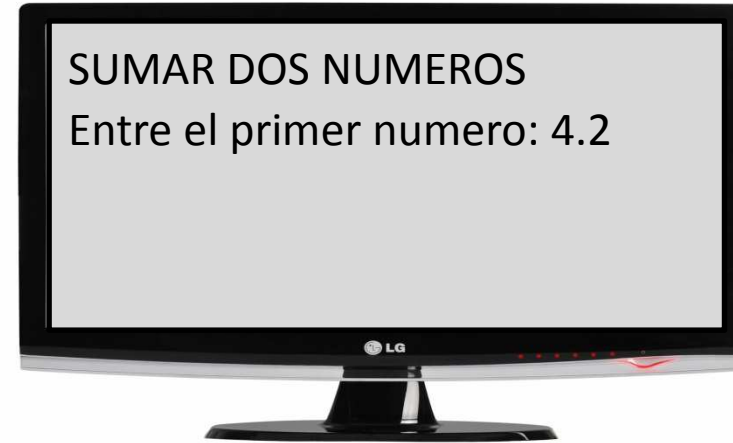
fin

# Resultado de 'ejecutar' el algoritmo del Ejemplo 1

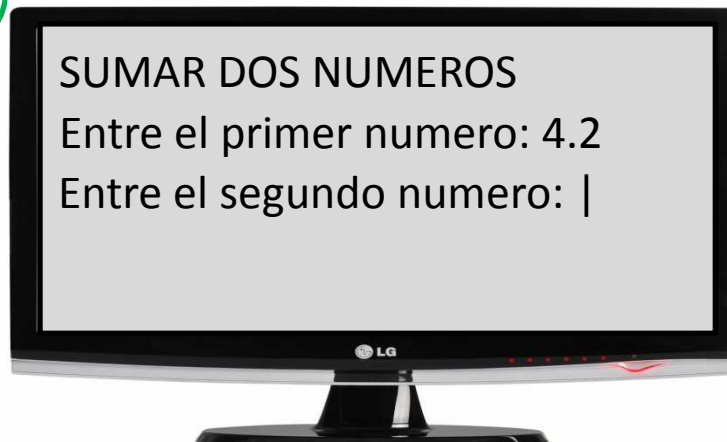
1



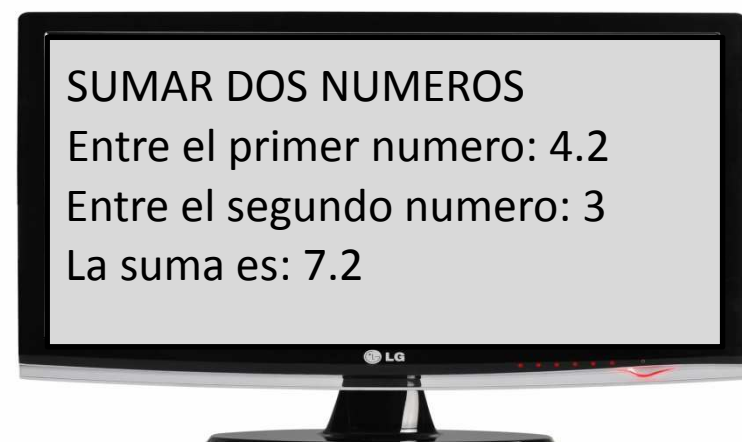
2



3



4



# Efectos de los pasos especificados en el algoritmo del Ejemplo 1

Cuando se ejecuta el algoritmo, en pantalla (dispositivo de salida) solo se ven los mensajes que se envíen con el procedimiento **muestre** y el computador espera a que se ingrese un valor cada vez que se ejecute el procedimiento **lea**. Las instrucciones que están entre las palabras reservadas **inicio** y **fin** producen los efectos mostrados en la figura anterior. A continuación se explican dichos efectos:

1. En el monitor se observan los mensajes "SUMAR DOS NUMEROS" y "Entre el primer numero: " porque el procedimiento **muestre** los envió. Solo se muestran esos dos mensajes porque luego se ejecuta el procedimiento **lea** para poder asignarle un valor a la variable a y dicho procedimiento hace que el computador espere hasta que se ingrese un valor y se presione la tecla Enter.
2. El usuario ingresa un valor numérico (4.2) y a continuación presiona la tecla Enter. De esta manera a la variable a se le asigna el numero 4.2
3. Se muestra el mensaje "Entre el segundo numero: " y el algoritmo se detiene hasta que el usuario ingrese un numero y presione la tecla Enter.
4. El usuario ingresa el numero 3 y a continuación presiona la tecla Enter. La variable b toma el valor de 3. Luego, el algoritmo calcula la suma de los valores almacenados en las variables a y b y asigna el resultado a la variable c. Observe que el cálculo de la suma y la asignación del resultado a la variable c no se percibe por la pantalla. El usuario desconoce el proceso mediante el cual se obtuvo el resultado, incluso desconoce el nombre de las variables. Por último, se muestra el mensaje que indica el resultado de la suma y termina el algoritmo.



# Ejemplo 2: Algoritmo para calcular el área de un triángulo

algoritmo AreaTriangulo

variables

real: base, altura, area

inicio

muestre('CALCULAR EL AREA DE UN TRIANGULO')

muestre('Entre la base: ')

lea(base)

muestre('Entre la altura: ')

lea(altura)

area = base \* altura / 2

muestre('El área es: ', area)

fin

# Ejemplo 3: Algoritmo para calcular el área y el perímetro de un círculo

algoritmo AreaPerimetroCirculo

variables

    real: radio, area, perimetro

inicio

    muestre('CALCULAR EL AREA Y PERIMETRO DE UN CIRCULO')

    muestre('Entre el radio: ')

    lea(radio)

$area = \pi * radio^2$

$perimetro = 2 * \pi * radio$

    muestre('El área es: ', area)

    muestre('El perímetro es: ', perimetro)

fin

## Ejemplo 4

Diseñe un algoritmo que determine el pago semanal de un obrero que recibe \$22000 por hora laborada dado que ha trabajado cierta cantidad de horas y tiene un descuento del 10% de lo devengado por concepto de seguridad social.

# Algoritmo para el Ejemplo 4

algoritmo PagoSemanalObrero

variables

real: horas, pago, pagoNeto, seguridadSocial

inicio

muestre('CALCULAR PAGO SEMANAL')

muestre('Entre las horas laboradas: ')

lea(horas)

pago = 22000 \* horas

seguridadSocial = 0.1 \* pago

pagoNeto = pago - seguridadSocial

muestre('El pago semanal es: ', pagoNeto)

fin

# Ejercicios

- Solucionar nuevamente los ejemplos vistos.
- Tomando como guía las presentaciones "Resultado de 'ejecutar' el algoritmo del Ejemplo 1" y "Efectos de los pasos especificados en el algoritmo del Ejemplo 1" haga el mismo análisis para los demás algoritmos de los ejemplos propuestos.
- Solucionar el taller 1.

# Referencias

- Joyanes Aguilar, Luis. Fundamentos de Programación: Algoritmos, estructura de datos y objetos. 4ª edición. Madrid: McGraw-Hill. 2008.