



## UNIDAD CURRICULAR: ALGORITMICA Y PROGRAMACIÓN

### UNIDAD I. ALGORITMOS Y PROGRAMAS

#### CONTENIDO:

Concepto de Algoritmos y Programas

Lenguaje algorítmico y de programación

Partes de un algoritmo

Características y elementos para construir un algoritmo

Formas de representar un algoritmo

Traza de un algoritmo

Ejercicios resueltos

Referencias Bibliográficas





## UNIDAD I

### ALGORITMOS Y PROGRAMAS

#### CONCEPTO DE ALGORITMOS Y PROGRAMAS

##### Algoritmos:

Son varias las definiciones de algoritmos:

“Es un método para resolver un problema”. (Joyanes, 2008)

Es un conjunto finito de instrucciones precisas que realizan una tarea, la cual, dado un estado inicial, culminará por arrojar un estado final reconocible. (<http://www.conocimientosweb.net/portal/article782.html>)

“Es un conjunto de acciones o secuencia de operaciones que ejecutadas en un determinado orden, resuelven el problema.” [http://www.edukanda.es/mediatecaweb/data/zip/1164/page\\_46.htm](http://www.edukanda.es/mediatecaweb/data/zip/1164/page_46.htm)

Básicamente todas las definiciones coinciden en que la finalidad del algoritmo es resolver un problema a través de la ejecución de un conjunto de pasos. De tal manera, que cualquier actividad que realizamos en nuestra vida cotidiana podríamos expresarla como un algoritmo. Por ejemplo una receta de cocina, instalación de un electrodoméstico, realización de una tarea doméstica o cualquier otra actividad de la vida diaria. Este tipo de algoritmo que es realizado por el ser humano se denomina **“algoritmo no computacional”**, mientras que el algoritmo que se realiza para ser ejecutado por el computador se denomina **“algoritmo computacional”**.

##### Ejemplo de algoritmo no computacional:

Algoritmo para realizar una limonada

Inicio

Tomar un recipiente

Agregar el agua al recipiente

Cortar los limones



Exprimir los limones

Agregar el jugo de los limones al recipiente

Agregar azúcar

Revolver y disfrutar la limonada

Fin

### **Ejemplo de un algoritmo computacional:**

Algoritmo que calcula la suma de dos números

Inicio

Obtener los valores del primer número y del segundo número

Sumar el primer número mas el segundo número

Mostrar el resultado de la suma de los dos números.

Fin

### **Programas:**

Cuando el algoritmo se debe procesar en una computadora, se ha de expresar en un formato que se denomina programa. Un programa se escribe en un lenguaje de programación y las operaciones que conducen a expresar un algoritmo en forma de programa se llama programación.

## **LENGUAJE ALGORÍTMICO Y DE PROGRAMACIÓN**

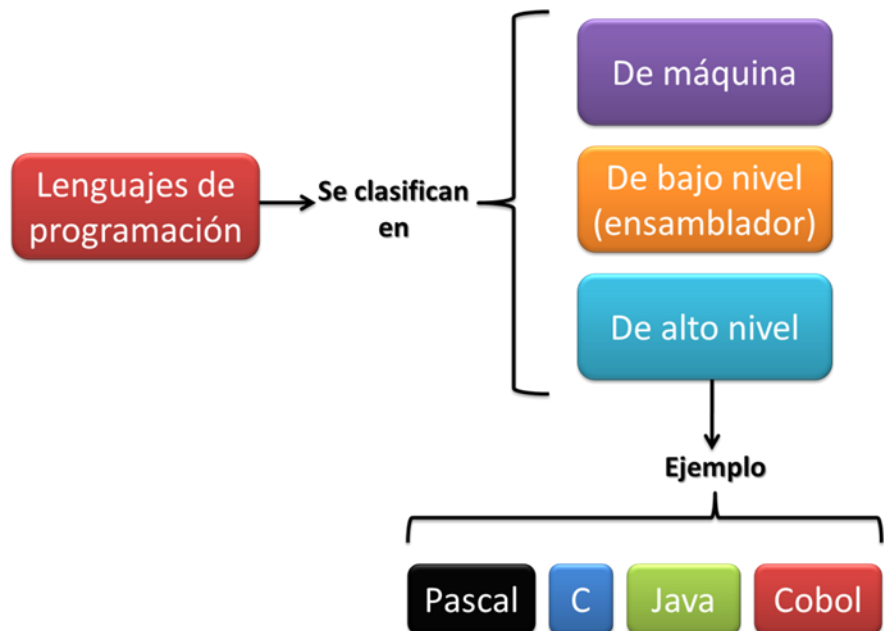
Lenguajes de programación son lenguajes utilizados para escribir programas de computadoras que permitan la comunicación usuario-máquina. Los principales tipos de lenguajes son:

**Lenguaje de máquina:** están escritos en código binario (0 y 1), y directamente son inteligibles por la máquina (computadora).



**Lenguaje de bajo nivel:** Es el lenguaje ensamblador basado en mnemotécnicos (abreviaturas) que no puede ser ejecutado directamente por la computadora sino que requiere una fase de traducción al lenguaje de máquina.

Los lenguajes ensambladores ofrecen una mayor facilidad de codificación que los lenguajes de máquina.



### Lenguaje de alto nivel:

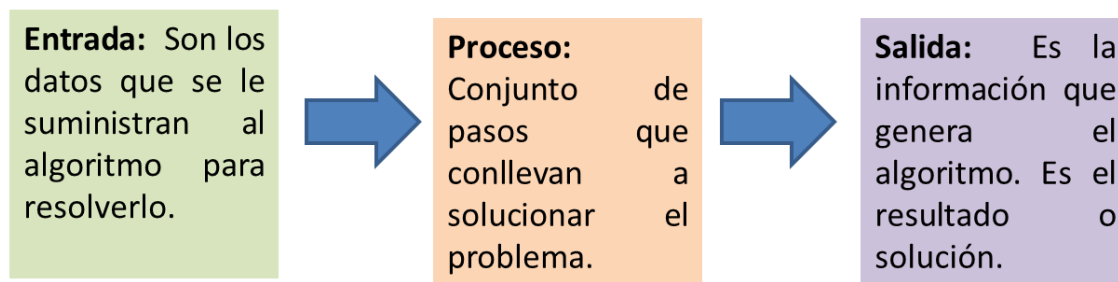
son los más utilizados por los programadores ya que su diseño se

aproxima mas al lenguaje natural humano lo que permite escribirlos y entenderlos de un modo mas fácil que los lenguajes de máquina o ensambladores. Ejemplos: Pascal, Basic, Gambas, Visual basic, php, phyton, Perl, Ruby, Java. C y C++ son lenguajes considerados de nivel medio ya que aún cuando poseen características de lenguajes de alto nivel también permiten ciertos manejos de bajo nivel.

## PARTES DE UN ALGORITMO

Un algoritmo consta de tres partes básicamente:

- **Entrada:** Son los datos necesarios que requiere el proceso para ofrecer los resultados esperados.
- **Proceso:** Son las operaciones básicas necesarias para obtener la solución del problema
- **Salida:** Resultados que se obtienen de las operaciones y representan la solución.



**Tomando en cuenta el ejemplo del Algoritmo para realizar la limonada, se pueden especificar las tres partes del algoritmo:**

**Entrada:** Los limones, el agua, el azúcar y el recipiente.

**Proceso:** elaboración de la limonada

**Salida:** La limonada.

**En el caso del algoritmo de calcular la suma de dos números, las tres partes serían:**

**Entrada:** Los dos números

**Proceso:** sumar los dos números

**Salida:** El resultado de la suma

## CARACTERÍSTICAS Y ELEMENTOS PARA CONSTRUIR UN ALGORITMO

Las características básicas que debe cumplir todo algoritmo son:

- Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
- Un algoritmo debe ser definido. Si se ejecuta un algoritmo dos veces, se debe obtener el mismo resultado cada vez.
- Un algoritmo debe ser finito. Si se ejecuta un algoritmo, se debe terminar en algún momento, es decir, debe tener un número finito de pasos.
- Un algoritmo debe ser correcto, debe hacer lo que se supone que hace generando la solución de manera correcta.



## FORMAS DE REPRESENTAR UN ALGORITMO: LENGUAJE NATURAL, PSEUDOCÓDIGO, Y DIAGRAMA DE FLUJO

Existen varias formas de representar un algoritmo: **lenguaje natural, pseudocódigo y diagrama de flujo.**

**El lenguaje natural:** es el medio que utilizamos de manera cotidiana para comunicarnos con las demás personas. Los algoritmos se pueden representar en lenguaje natural descomponiéndolo en una serie de pasos secuenciales descriptivos, tal como se realizó en los ejercicios anteriores, sin embargo, esta forma suele ser ambigua y extensa, por lo tanto, se recomienda usar pseudocódigo y diagramas de flujo para reducir esta ambigüedad y se facilite su conversión a programas de computadora.

**Pseudocódigo:** Es una herramienta de programación en la que las instrucciones se escriben en palabras en inglés o español, que facilitan tanto la escritura como la lectura de programas. En otras palabras, es un lenguaje de especificación (descripción) de algoritmos, formado por una serie de palabras que representan los pasos del algoritmo. Estas palabras son traducción libre de palabras reservadas de lenguajes de programación como Basic, Pascal o C. El pseudocódigo no puede ser ejecutado por una computadora pero es muy útil para el programador ya que podrá centrar su atención en la lógica y en las estructuras de control sin preocuparse de las reglas de un lenguaje específico.

Dado que no existe un único pseudocódigo para representar algoritmos, para efectos del presente material educativo computarizado se utilizará las siguientes palabras para el pseudocódigo en las próximas unidades temáticas: ***Inicio, fin, leer, escribir, si- entonces -de lo contrario, haga mientras-fin del haga mientras, haga desde-fin del haga desde, haga- hasta.***

La escritura del pseudocódigo debe tener indentación (sangría) para su mejor entendimiento. Se comienza con el nombre del algoritmo y la palabra ***Inicio***, culmina con la palabra ***Fin***.

Los datos de entrada deben ir precedidos por la palabra ***Leer***



La salida o solución debe ir precedida por la palabra **Escribir**

El proceso se representa por cálculos, fórmulas, soluciones.

### Ejemplo:

Algoritmo SUMAR

Inicio

Leer num1, num2




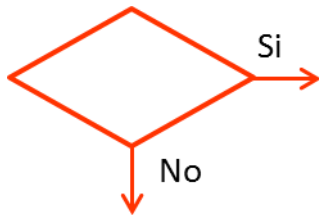
Suma=num1+num2

Escribir Suma

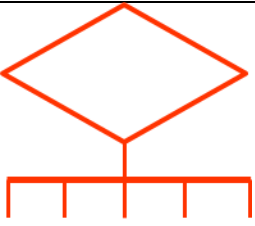
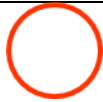








Fin

### Diagrama de flujo:

Es una representación gráfica de un algoritmo a través de diagramas que utiliza símbolos y que tiene los pasos del algoritmo escritos en dichos símbolos unidos por flechas, denominadas líneas de flujo que indican la secuencia en que se deben ejecutar.

| Símbolo   | Función   |
|---|---|
|  | <b>Terminal:</b> Representa “inicio” y “fin” de un programa.  |
|  | <b>Entrada/Salida:</b> Representa cualquier tipo de entrada o salida de datos.  |
|  | <b>Proceso:</b> Cualquier tipo de operación, cambio de valor, operaciones aritméticas, entre otras.   |
|  | <b>Decisión:</b> Indica operaciones lógicas o de comparación entre datos. En función del resultado se determina cuál de los distintos caminos alternativos del programa se debe seguir. Normalmente tiene dos salidas: SI o NO, pero puede tener tres o más, según sean los |

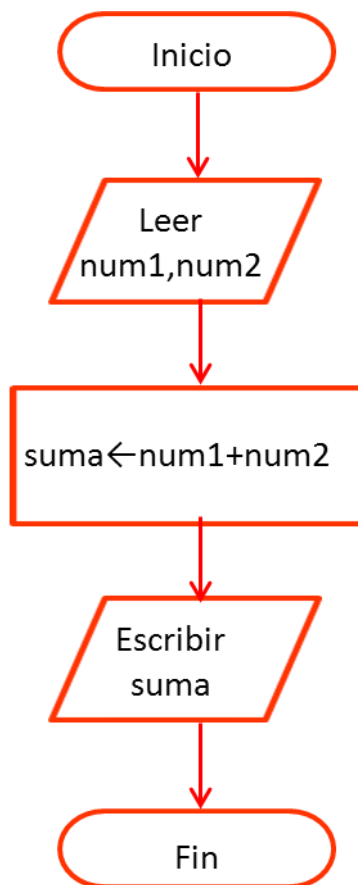


|   |   |
|---|---|
|   | <b>casos</b>  |
|    | <b>Decisión múltiple:</b> en función del resultado de la comparación se seguirá uno de los diferentes caminos de acuerdo a dicho resultado. |
|    | <b>Conector:</b> se utiliza para conectar caminos en la misma página del diagrama   |
|    | <b>Indicador de dirección:</b> indica el flujo secuencial de la lógica del programa   |
|    | <b>Línea conectora:</b> sirve de unión entre dos símbolos   |
|    | <b>Conector:</b> conecta dos puntos del diagrama situado en páginas diferentes  |
|   | Llamada a subrutina o a un proceso predeterminado.  |
| <b>Símbolos Secundarios</b>   | <b>Función</b>  |
|  | <b>Pantalla:</b> salida por pantalla  |
|  | <b>Impresora:</b> Salida por impresora  |
|  | <b>Teclado:</b> entrada por teclado   |
|  | <b>Comentarios:</b> permite añadir comentarios a otros símbolos del diagrama de flujo. Se puede dibujar a cualquier lado del símbolo        |





### Ejemplo de un diagrama de flujo para calcular la suma de dos números:



### TRAZA DE UN ALGORITMO (CORRIDA EN FRÍO)

La traza de un algoritmo o también llamada corrida, o corrida en frío del algoritmo indica la secuencia de pasos o instrucciones tal como se ejecutarían en un computador, asignando valores a las variables y registrando el comportamiento de las mismas después de cada acción. Es una técnica muy útil para comprobar si el algoritmo genera una solución satisfactoria al problema planteado.



## EJERCICIOS RESUELTOS

### Algoritmos no computacionales

#### a) Algoritmo para comenzar a trabajar en el computador

Inicio

Conectar el cable del regulador en el enchufe.

Prender el regulador.

Oprimir el botón de encendido del computador.

Oprimir el botón de encendido del monitor.

Esperar que inicie el sistema operativo.

Comenzar a trabajar en el computador

Fin

#### b) Algoritmo para cambiar un caucho espichado

Inicio

Aflojar los tornillos del caucho espichado con la llave inglesa.

Ubicar el gato mecánico en el sitio respectivo.

Levantar el gato hasta que la rueda pinchada pueda girar libremente. Quitar los tornillos y el caucho espichado.

Poner el caucho de repuesto y los tornillos.

Bajar el gato.

Sacar el gato de su sitio.

Apretar los tornillos con la llave inglesa.

Fin

#### c) Algoritmo para realizar una llamada telefónica

Inicio

Descolgar el auricular

Esperar el tono

Marcar el número

Esperar que contesten

Hablar

Colgar el auricular



Fin

### **Algoritmos Computacionales**

#### **d) Algoritmo para convertir horas a minutos**

Inicio

Leer horas

$\text{minutos} \leftarrow \text{horas} * 60$

Escribir minutos

Fin

#### **e) Algoritmo para determinar el promedio de 4 calificaciones**

Inicio

Leer nota1, nota2, nota3, nota4

$\text{promedio} = (\text{nota1} + \text{nota2} + \text{nota3} + \text{nota4}) / 4$

Escribir promedio

Fin

#### **f) Algoritmo para calcular el área de un cuadrado**

Leer lado

$\text{area} = \text{lado} * \text{lado}$

Escribir área

Fin



## REFERENCIAS BIBLIOGRÁFICAS

Bassard, G y Bratley, P. (2010). Fundamentos de algoritmia. Prentice-Hall.

ConocimientosWeb.net. La divisa del nuevo milenio. Introducción a los algoritmos. [Página web en línea]. Disponible: <http://www.conocimientosweb.net/portal/article782.html>. Consulta: 2014, Mayo 3.

Guadalinfo.es. La ciudad virtual más grande de Andalucía. Crea aplicaciones para tu móvil. Introducción a la programación. [Página web en línea]. Disponible: [http://www.edukanda.es/mediatecaweb/data/zip/1164/page\\_46.htm](http://www.edukanda.es/mediatecaweb/data/zip/1164/page_46.htm). Consulta: 2014, Mayo 3.

Joyanes, L. (2008). Fundamentos de programación. Algoritmos , Estructuras de datos y objetos. Mc Graw Hill. Tercera edición.