

	FACULTAD DE INGENIERÍAS ÁREA DE PROGRAMAS INFORMÁTICOS	Guía # 3 Versión: 01
---	---	-------------------------

ASIGNATURA: Algoritmos y Programación 1	CÓDIGO: ING01179
DOCENTE: Luis Fernando González Alvarán	FECHA: abril 29/2022

SEMINARIO DE ENCUENTRO NÚMERO 3

PRESEMINARIO # 3

El estudiante debe realizar la siguiente lectura y complementarla con la bibliografía, o los enlaces sugeridos, buscando temas relacionados con la unidad en estudio, posteriormente realizará una socialización de lo estudiado, haciendo sus aportes o manifestando las inquietudes, en el chat de la asignatura.

I. METODOLOGÍA PARA LA SOLUCIÓN DE PROBLEMAS POR COMPUTADOR.

Existen una serie de principios que gobiernan el comportamiento de los computadores y programas; los cuales establecen la forma correcta de abordar un problema cuando se pretende resolverlo a partir de un computador. Este conjunto de principios recibe el nombre de Resolución de problemas algorítmicos.

En consecuencia, de lo anterior se hace necesario fundamentar y adquirir un conocimiento firme sobre lo que es un algoritmo, y lo que es un problema algorítmico; y así poder establecer claramente la forma de resolver dichos problemas.

1. CONCEPTO DE ALGORITMO.

Un algoritmo es un método para resolver un problema, aunque formalmente se define como un conjunto de instrucciones, procedimientos o acciones que realizan una descripción paso a paso y precisa de cierto proceso; garantizando alcanzar un resultado o resolver un problema determinado.

Aunque la popularización del termino ha llegado con el advenimiento de la era informática, algoritmo proviene de Mohammed al-KhoWârizmi, matemático persa que vivió durante el siglo IX y alcanzó gran reputación por el enunciado de las reglas paso a paso para sumar, restar, multiplicar y dividir números decimales; la traducción al latín del apellido es la palabra algorismus la cual derivó posteriormente en algoritmo. Euclides, el gran matemático griego (del siglo IV a.C.) que inventó un método para encontrar el máximo común divisor de dos números, se considera con Al-Khowarizmi el otro gran padre de la algoritmia (ciencia que trata de los algoritmos).

	FACULTAD DE INGENIERÍAS ÁREA DE PROGRAMAS INFORMÁTICOS	Guía # 3 Versión: 01
---	---	-------------------------

El profesor Niklaus Wirth —inventor de Pascal, Modula-2 y Oberon— tituló uno de sus más famosos libros, Algoritmos + Estructuras de datos = Programas, indicando que solo se puede llegar a realizar un buen programa con el diseño de un algoritmo y una correcta estructura de datos. Esta ecuación será una de las hipótesis fundamentales consideradas en este libro.

Para llegar a la realización de un programa es necesario el diseño previo de un algoritmo, de modo que sin algoritmo no puede existir un programa.

Los algoritmos son independientes tanto del lenguaje de programación en que se expresan como del computador que los ejecuta. En cada problema el algoritmo se puede expresar en un lenguaje diferente de programación y ejecutarse en un computador distinto; sin embargo, el algoritmo será siempre el mismo. Así, por ejemplo, en una analogía con la vida diaria, una receta de un plato de cocina se puede expresar en español, inglés o francés, pero cualquiera que sea el lenguaje, los pasos para la elaboración del plato se realizarán sin importar el idioma del cocinero.

En la ciencia de la computación y en la programación, los algoritmos son más importantes que los lenguajes de programación o los computadores. Un lenguaje de programación es tan solo un medio para expresar un algoritmo y un computador es solo un procesador para ejecutarlo. Tanto el lenguaje de programación como el computador son los medios para obtener un fin: conseguir que el algoritmo se ejecute y se efectúe el proceso correspondiente.

Dada la importancia del algoritmo en la ciencia de la computación, un aspecto muy importante será el diseño de algoritmos.

El diseño de la mayoría de los algoritmos requiere creatividad y conocimientos de las técnicas de programación. En esencia, la solución de un problema se puede expresar mediante un algoritmo.

Muchas veces se aplican algoritmos de manera inadvertida, inconsciente o automáticamente. Esto generalmente se produce cuando el problema al que nos enfrentamos se ha resuelto con anterioridad varias veces.

1.1. CARACTERÍSTICAS DE LOS ALGORITMOS.

Las características fundamentales que debe cumplir todo algoritmo son:

- Debe tener un punto particular de inicio.
- Un algoritmo debe ser preciso e indicar el orden de realización de cada paso.
- Un algoritmo debe estar definido. Si se sigue un algoritmo dos veces, se debe obtener el mismo resultado cada vez. No debe permitir dobles interpretaciones.
- Un algoritmo debe ser finito. Si se sigue un algoritmo, se debe terminar en algún momento; o sea, debe tener un número finito de pasos. Debe ser finito en tamaño y tiempo de ejecución.
- Debe ser general, es decir, soportar la mayoría de las variantes que se puedan presentar en la definición del problema.

La definición de un algoritmo debe incluir tres partes: Entrada, Proceso y Salida.

En el ejemplo de una receta de cocina, del cual se habló anteriormente se pueden identificar:

Entrada: Ingredientes y utensilios empleados.
Proceso: Elaboración de la receta en la cocina.
Salida: Terminación del plato (por ejemplo, "Pato a la Naranja")

La resolución de un problema exige el diseño de un algoritmo que resuelva el problema propuesto.

El programador de computadores es antes que nada una persona que resuelve problemas, por lo que para llegar a ser un programador eficaz se necesita aprender a resolver problemas de un modo riguroso y sistemático. A lo largo de todo este libro se hablará de la metodología necesaria para resolver problemas mediante programas, concepto que se denomina metodología de la programación. El eje central de esta metodología es el concepto de algoritmo.

1.2. PROBLEMA ALGORÍTMICO.

Un problema algorítmico es cualquier problema conceptual o práctico cuya solución puede expresarse mediante un algoritmo.

A diario se encuentran muchos problemas algorítmicos, no solamente en el campo de la informática, es decir, muchas de las acciones rutinarias de una persona, obedecen a un problema algorítmico.

Ejemplo.

PROBLEMA ALGORÍTMICO	ALGORITMO
Dar la vuelta a oriente	Itinerario particular del recorrido.
Cambiar una llanta estallada	Conjunto de pasos para el cambio de la llanta

Es importante tener en cuenta que no todos los algoritmos pueden ser ejecutados por un computador, pues los computadores solo pueden ejecutar algoritmos que se componen de acciones individuales que se pueden entender y realizar, por ejemplo la preparación del pollo a la cazadora implica acciones como "deshuesar el pollo", tarea para la que un computador aún no está preparado. Por lo anterior es necesario conocer bien cuáles son las tareas que puede realizar un computador, de forma que se diseñen algoritmos que contengan solo este tipo de tareas, obedeciendo al modelo computacional que se plantea en la figura 1. y que es congruente con las partes de un algoritmo.

Figura 1. Modelo de computación



Este modelo plantea que cualquier proceso de cálculo se compone de tres partes: una

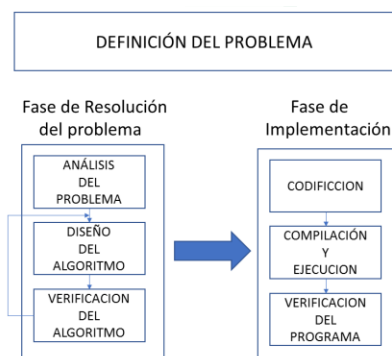
entrada un proceso y una salida; donde **la entrada** se constituye por un conjunto de información que necesitan los pasos de los cuales se compone el algoritmo para llevar a cabo la tarea; **el proceso** contiene (una descripción de) los pasos del algoritmo; y finalmente **la salida** constituye el resultado que se obtiene ejecutando los pasos con los datos de entrada.

Por ejemplo el procedimiento para calcular la calificación media es un algoritmo que si se puede ejecutar en un computador, donde las entradas corresponden a una lista de valores numéricos dentro del rango 0 a 5; la salida será la media de esos valores; y el proceso es (una descripción de) el conjunto de pasos individuales que deben seguirse para obtener el resultado.

Para llegar a identificar claramente las entradas, procesos y salidas de un algoritmo e implementarlas en un computador se ha definido una metodología la cual se describe a continuación.

II. RESOLUCION DE PROBLEMAS POR COMPUTADOR.

La solución de un problema algorítmico con la ayuda del computador recibe el nombre de resolución de problemas por computador y es una metodología que consta de dos fases cada una compuesta de ciertas etapas y que todas en conjunto conducen a la escritura de un programa y a la ejecución de éste; pero para poder abordar el proceso detalladamente se hace necesario cumplir con una fase preliminar que no hace parte del proceso de resolución como tal, aunque muchos autores la incluyen en éste y que es la definición del problema, ésta apunta al entendimiento claro del problema. Un problema no se puede pretender solucionar, sin tener un claro entendimiento de éste y lo que se espera en su solución.



1.1. DEFINICIÓN DEL PROBLEMA: Esta fase está dada por el enunciado del problema, el cual requiere una definición clara y precisa. Es importante que se conozca lo que se desea que realice el computador; mientras esto no se conozca del todo no tiene mucho caso continuar con el proceso. La definición del problema va estrechamente ligada con el entendimiento detallado de éste. No es posible definir con certeza una solución, si no se tiene claridad acerca de la dimensión del problema.

Las etapas para la resolución de un problema por computador se agrupan en dos grandes fases:

1.2. FASE RESOLUCIÓN DEL PROBLEMA: Esta corresponde a todas las actividades previas al uso del computador en la solución de un problema. Se compone de las siguientes actividades:

1.2.1. Análisis del Problema: En esta etapa se hace una clara interpretación del problema, donde se contempla exactamente lo que debe hacer el programa y el resultado o solución deseada. Una vez que se ha comprendido lo que se desea del computador, es necesario definir:

1.2.1.1. Modelamiento: Obedece a la identificación y descripción de los datos de entrada, salida y auxiliares; cada uno de estos asociados a un respectivo identificador.

Es de aclarar que no necesariamente tienen que existir los datos auxiliares, e incluso en ciertas ocasiones puede suceder con los de entrada.

- **Datos de entrada.** Corresponde a los datos que se requieren para la solución del problema. Son aquellos que no se conocen y que no existe forma de deducirlos en el transcurso de la solución.
- **Datos Auxiliares.** Son los datos que son conocidos desde el enunciado del problema, o que pueden obtenerse a partir de los datos de entrada y una información adicional, pero que no obedecen a los resultados esperados. Sirven para el logro de la solución.
- **Datos de Salida.** Obedecen a la información que se desea producir como resultado, es decir corresponden a la solución del problema planteado.

Una recomendación muy práctica, es que a la hora de analizar el problema planteado, la persona que analiza, se coloque en el lugar del computador y piense que es lo que se necesita que se le ordene y en que secuencia para producir los resultados esperados. Este ejercicio ayuda a identificar los datos de entrada, salida y auxiliares

Ejemplo: Dado un número positivo obtener el resultado de elevar al cuadrado el producto de dicho número multiplicado por 3.

DATO DE ENTRADA

Descripción

Numero dado

Identificador

Num

DATOS AUXILIAR

Descripción

Producto del número por 3

Identificador

Producto

DATO DE SALIDA

Descripción

Cuadrado del producto

Identificador

Cu_del_P

1.2.1.2. Especificaciones: Obedece a un análisis intuitivo del estado de un algoritmo, específicamente de los diferentes datos que van a hacer parte de la solución, donde

se describen como están éstos antes y después de que se haya ejecutado un paso del algoritmo. Existen dos pasos de interés especial: el estado inicial que recibe el nombre de *precondición* del algoritmo (o pre); y el estado final que recibe el nombre de *postcondición* del algoritmo (o post)

- **Precondiciones.** Corresponde a una descripción del estado inicial de las entradas, es decir antes de que se ejecute el primer paso del algoritmo; por lo tanto, corresponde a las especificaciones de los identificadores de entrada. En esta descripción se indica el tipo de dato de cada una de las variables y el valor de las constantes. Además, en el caso de las variables se deben establecer los límites de los datos, factor que posteriormente determinará las condiciones de validación
- **Postcondiciones.** Corresponde a la descripción de las variables auxiliares y de salida, definiendo claramente el tipo de dato o condiciones específicas de los resultados.

Ejemplo: (basado en el modelamiento anterior)

Precondiciones

Num $\in \{\text{Reales} > 0\}$

Postcondiciones

Producto $\in \{\text{Reales}\}$

Cu_del_P $\in \{\text{Reales}\}$

1.2.1.3. Procesos y/o formulas: Corresponde a la definición de métodos y/o fórmulas que se necesitan utilizar para procesar los datos y llegar a obtener el valor de las variables auxiliares y de salida, se deben plantear en estricto orden de uso.

Ejemplo (basado en el anterior)

Fórmulas

Producto $\leftarrow \text{Num} * 3$

Cu_del_P $\leftarrow \text{Producto} ^ 2$

El estricto orden de planteamiento se refiere a que (según el ejemplo) no se puede plantear el proceso de **Cu_del_P** sin previamente haber definido a **Producto**, ya que la primera implica haber calculado la segunda.

1.2.2. Diseño del Algoritmo: Una vez se tenga claridad del problema, el modelamiento, las especificaciones y los procesos y/o fórmulas; se diseña una solución que obedece a un algoritmo que resuelva el problema, el cual consiste en hacer una descripción de los pasos lógicos que dan solución al problema. Dicha solución se puede realizar de forma gráfica (diagrama) o no grafica (pseudocódigo). *Estas formas se ven con detalle en el preseminario siguiente.*

1.2.3. Verificación del algoritmo: Cuando se crea un algoritmo es posible que se cometan errores de tipo lógico; estos errores no pueden ser detectados por el

	FACULTAD DE INGENIERÍAS ÁREA DE PROGRAMAS INFORMÁTICOS	Guía # 3 Versión: 01
---	---	-------------------------

computador, es decir, los resultados que éste arroje simplemente son errados. El proceso de verificación permite detectar los posibles errores que se cometan en el diseño de un algoritmo y así corregirlos antes de continuar con el proceso. Esta verificación se conoce como prueba de escritorio y corresponde a un seguimiento de las secuencias del algoritmo, a partir de datos extremos. Los resultados obtenidos se analizan comparativamente con resultados reales y así se puede establecer el correcto funcionamiento o no del algoritmo. En caso de encontrarse inconsistencias, se procede a la búsqueda y corrección del error.

1.3. FASE DE IMPLEMENTACIÓN.

1.3.1. Codificación: La solución sin errores lógicos se escribe en un lenguaje de programación de alto nivel.

Simplemente se convierten uno a uno los pasos del algoritmo a instrucciones del lenguaje seleccionado, lo anterior se realiza en un editor de textos o en el propio compilador del lenguaje; cumpliendo con la sintaxis específica de éste. El resultado de este proceso recibe el nombre de programa fuente.

1.3.2. Compilación y ejecución: La compilación es un proceso de traducción de programas fuentes a programas objeto, durante el cual se realiza el análisis de cada una de las instrucciones escritas en el lenguaje de programación seleccionado, acompañado de una rigurosa detección de errores de sintaxis (denominados bugs). El programa resultado después de la compilación objeto corresponde a un código de máquina que no posee errores y el cual será ejecutado.

1.3.3. Verificación del Programa: Esta fase está estrechamente ligada con la anterior y corresponde a un proceso de pruebas y refinamiento del código, acorde con un periodo donde se analizan los resultados obtenidos y se comparan con resultados reales.