

Contenido

1. Programación competitiva
2. Algoritmos y resolución de problemas
3. Programación y Lenguajes de programación

Programación Competitiva

¿Qué es la programación competitiva?

La programación competitiva se podría resumir como “Dado un conjunto de bien conocidos problemas de las Ciencias de la Computación, resolverlos lo más rápido y eficientemente posible”. Esta definición nos lleva a conceptualizar que la programación competitiva consiste en: Competir por ser el mejor en la resolución de problemas de la Ciencias de la Computación. En ella prima el trabajo en equipo, el análisis de problemas y el desarrollo rápido de soluciones.

La programación competitiva surge a inicios de los años 70, cuando un grupo de estudiantes de la Universidad A&M de Texas, Estados Unidos, deciden hacer una competencia para saber quién de ellos era el más eficiente programador. Siete años después esta competencia se convirtió en un evento Inter universidades con varias rondas clasificatorias. Actualmente la Internacional ACM-ICPC es la principal promotora de estos eventos. Con presencia en más de 90 países, la participación de más de 1800 universidades y un estimado de 20000 participantes ACM-ICPC congrega cada año estudiantes de todo mundo en un campeonato mundial cuyo fin principal es develar al mejor programador.

En el Perú, grupos no gubernamentales y sociedades privadas como la SPC (Sociedad Peruana de Computación), la CPC (Concurso Peruano de Programación) en coordinaciones con la ACM-ICPC Perú han venido promoviendo estas actividades hace algunos años. Estudiantes universitarios de nuestro país han encontrado en estos eventos nuevas oportunidades de desarrollo intelectual y profesional, un ejemplo claro fue cuando un equipo peruano clasificó por primera vez al campeonato mundial del año 2011 realizado entonces en Orlando, Estados Unidos quedando entre los 100 mejores equipos del mundo. El campeonato mundial de la ACM-ICPC ha tenido presencia de equipos peruanos también en el año 2012 y 2013.

Algoritmos y Resolución de problemas

¿Qué es un algoritmo?

Por algoritmo, se entiende a un conjunto finito de instrucciones que se deben seguir para resolver un problema. No obstante, desde el punto de vista de la programación de ordenadores, la definición del algoritmo como la especificación de una serie de pasos, es incompleta. Debe observarse que los ordenadores son equipos que tienen limitaciones físicas en cuanto a capacidad de almacenamiento y procesamiento. Por consiguiente debemos refinar un poco más nuestra definición de algoritmo para hacerla aplicable de manera efectiva en el ámbito de la informática.

El algoritmo es un conjunto de pasos, instrucciones o acciones que se deben seguir para resolver un problema. Existen una gran cantidad de algoritmos, hay que coger el más efectivo. Hay dos tipos de algoritmos que son los cualitativos y cuantitativos, cualitativos son todos aquellos pasos o instrucciones descritos por medio de palabras que sirven para llegar a la obtención de una respuesta o solución de un problema, y cuantitativos son todos aquellos pasos o instrucciones que involucran cálculos numéricos para llegar a un resultado satisfactorio.

Características:

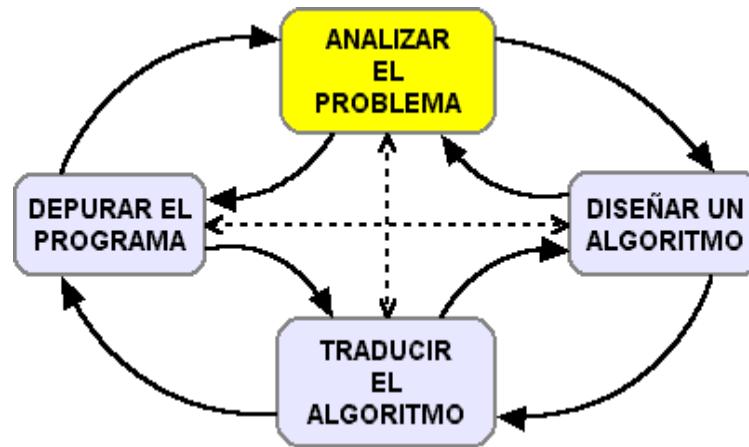
- Tiene que ser preciso.
- Tiene que estar bien definido.
- Tiene que ser finito.
- La programación es adaptar el algoritmo al ordenador.
- El algoritmo es independiente según donde lo implemente.

Análisis de problemas

Los programas de computador tienen como finalidad resolver problemas específicos y el primer paso consiste en definir con precisión el problema hasta lograr la mejor comprensión posible. Una forma de realizar esta actividad se basa en formular claramente el problema, especificar los resultados que se desean obtener, identificar la información disponible (datos), determinar las restricciones y definir los procesos necesarios para convertir los datos disponibles (materia prima) en la información requerida (resultados).

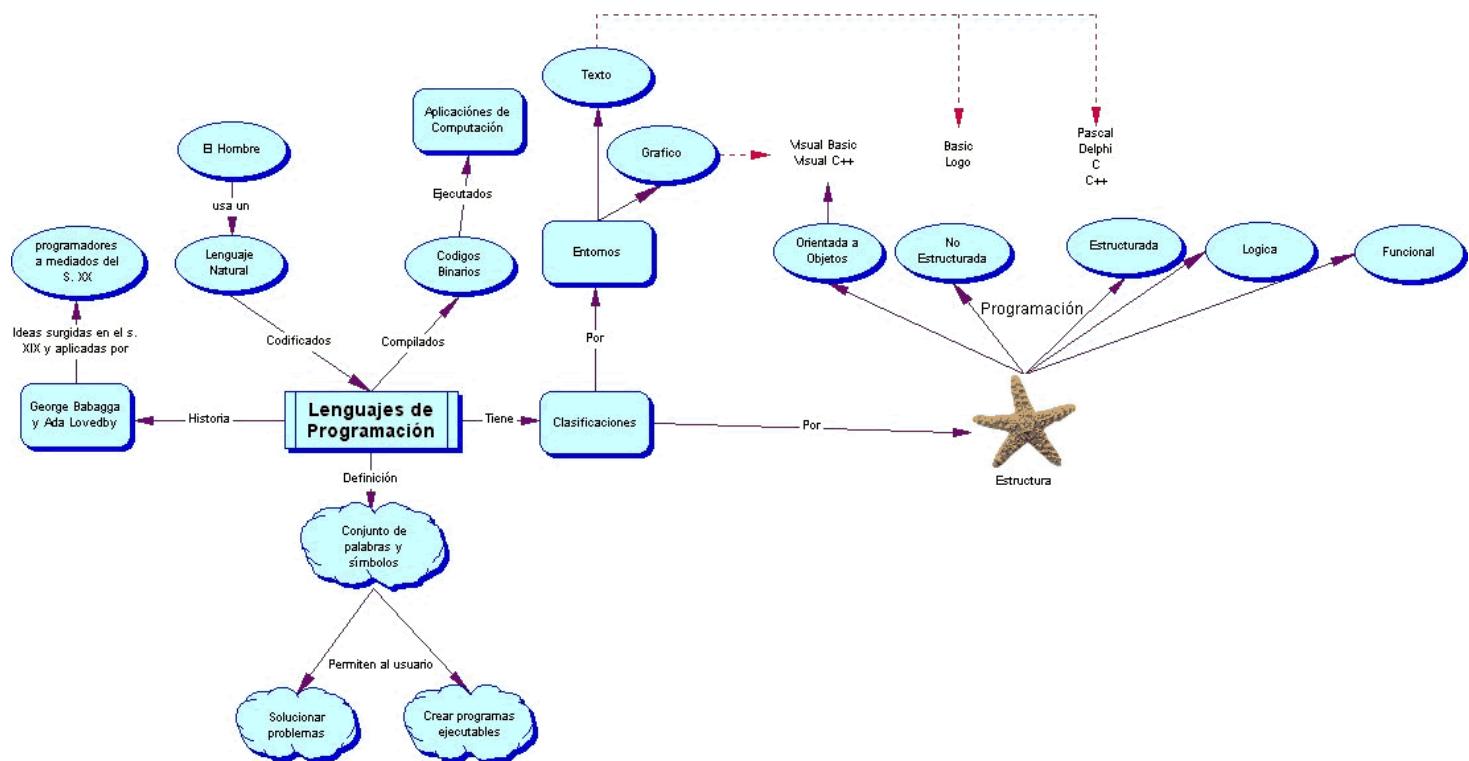
Estas etapas coinciden parcialmente con los elementos generales que deben estar presentes en todos los problemas:

1. Especificar claramente los resultados que se desean obtener (*meta y sub metas*)
2. Identificar la información disponible (*estado inicial*)
3. Definir los procesos que llevan desde los datos disponibles hasta el resultado deseado (*operaciones*)



Programación y Lenguajes de Programación

Un lenguaje de programación (LP) es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana. Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación.



Elementos de un Lenguaje de Programación

Variables y vectores

Las variables podrían calificarse como contenedores de datos y por ello se diferencian según el tipo de dato que es capaz de almacenar. En la mayoría de lenguajes de programación se requiere especificar un tipo de variable concreto para guardar un dato concreto. Por ejemplo, en **JAVA**, si deseamos guardar una cadena de texto deberemos especificar que la variable es del tipo **String**. Por otra parte, en lenguajes como el **PHP** este tipo de especificación de variables no es necesario.

Además, existen variables compuestas por varias variables llamadas vectores. Un vector no es más que un conjunto de variables consecutivas en memoria y del mismo tipo guardadas dentro de una variable contenedor. A continuación, un listado con los tipos de variables y vectores más comunes:

- **Variables tipo char:** Estas variables contienen un único carácter, es decir, una letra, un signo o un número.
- **Variables tipo int:** Contienen un número entero.
- **Variables tipo float:** Contienen un número decimal.
- **Variables tipo String:** Contienen cadenas de texto, o lo que es lo mismo, es un vector con varias variables del tipo Char.
- **Variables del tipo boolean:** Solo pueden contener un 0 o un 1. El cero es considerado para muchos lenguajes como el literal "**false**", mientras que el 1 se considera "**true**".

Condicionantes

Los condicionantes son estructuras de código que indican que, para que cierta parte del programa se ejecute, deben cumplirse ciertas premisas; por ejemplo: que dos valores sean iguales, que un valor exista, que un valor sea mayor que otro... Estos condicionantes por lo general solo se ejecutan una vez a lo largo del programa. Los condicionantes más conocidos y empleados en programación son:

- **if:** Indica una condición para que se ejecute una parte del programa.
- **else if:** Siempre va precedido de un "If" e indica una condición para que se ejecute una parte del programa siempre que no cumpla la condición del if previo y si se cumpla con la que el "else if" especifique.
- **else:** Siempre precedido de "If" y en ocasiones de "else If". Indica que debe ejecutarse cuando no se cumplan las condiciones previas.

Bucles

Los bucles son parientes cercanos de los condicionantes, pero ejecutan constantemente un código mientras se cumpla una determinada condición. Los más frecuentes son:

- **for:** Ejecuta un código mientras una variable se encuentre entre 2 determinados parámetros.
- **while:** Ejecuta un código mientras se cumpla la condición que solicita.

Hay que decir que a pesar de que existan distintos tipos de bucles, ambos son capaces de realizar exactamente las mismas funciones. El empleo de uno u otro depende, por lo general, del gusto del programador.

Funciones

Las funciones se crearon para evitar tener que repetir constantemente fragmentos de código. Una función podría considerarse como una variable que encierra código dentro de si. Por lo tanto cuando accedemos a dicha variable (la función) en realidad lo que estamos es diciendo al programa que ejecute un determinado código predefinido anteriormente.

Todos los lenguajes de programación tienen algunos elementos de formación primitivos para la descripción de los datos y de los procesos o transformaciones aplicadas a estos datos (tal como la suma de dos números o la selección de un elemento que forma parte de una colección). Estos elementos primitivos son definidos por reglas sintácticas y semánticas que describen su estructura y significado respectivamente.

Sintaxis

A la forma visible de un lenguaje de programación se le conoce como sintaxis. La mayoría de los lenguajes de programación son puramente textuales, es decir, utilizan secuencias de texto que incluyen palabras, números y puntuación, de manera similar a los lenguajes naturales escritos.

Por otra parte, hay algunos lenguajes de programación que son más gráficos en su naturaleza, utilizando relaciones visuales entre símbolos para especificar un programa. Con frecuencia se resaltan los elementos de la sintaxis con colores diferentes para facilitar su lectura. Este ejemplo está escrito en **JAVA**.

```
2
3  public class Main {
4
5      public static void main(String[] args) {
6
7          int numero = 31;
8          double decimal = 3.5;
9          String texto = "Hello CoderPeru";
10         int[] array = {1,2,3};
11         System.out.println(texto);
12     }
13 }
14 }
15 }
```

La sintaxis de un lenguaje de programación describe las combinaciones posibles de los símbolos que forman un programa sintácticamente correcto. El significado que se le da a una combinación de símbolos es manejado por su semántica (ya sea formal o como parte del **CÓDIGO DURO** de la referencia de implementación).

Dado que la mayoría de los lenguajes son textuales, nos referimos a la **SINTAXIS TEXTUAL**.

NOTA. - Si usted no está familiarizado aun con **JAVA** o algún otro lenguaje de programación, solo le bastará unas pocas lecciones que puede encontrar en artículos, videos, manuales entre otros contenidos de internet para comenzar sus actividades en nuestra web, ¡Ojo! **Usted no necesita ser un experto programador para participar de nuestras actividades.**