

PROGRAMACIÓN

Unidad 2: Lenguajes de programación.
Programación estructurada. El lenguaje C.
Elementos básicos de un programa. Análisis de
programas. Estilo de programación



Recordemos algunos conceptos vistos
anteriormente...

Etapas en la resolución de problemas con computadora

- Definición y análisis del problema.
- Diseño del algoritmo.
- Codificación.
- Corrección, prueba y optimización.
- Verificación.
- Documentación.

Codificación - Escritura de programas

- El algoritmo se “traduce” al lenguaje de programación elegido.
- Escribir las instrucciones reales.
- Programar NO ES LO MISMO QUE codificar.



Corrección, prueba y optimización

Para la ejecución → traducción → lenguaje de maquina

En la compilación y prueba:

Los errores de ejecución.

Los errores de tipo lógico.

Los errores de sintaxis.

Verificación

Compilación → errores y advertencias.

Es necesario realizar las correcciones sintácticas y lógicas



Resultados deseados

Conjuntos de datos de prueba → Correctitud

Documentación

Documentación interna:

- Tabulación en el código dependiente.
- Uso de comentarios.

Documentación externa:

- Para el usuario.
- Para el programador.

Algunas recomendaciones

- La metodología Top-down.
- Divide and conquer
- Recursos del lenguaje
- Elección del tipo de dato apropiado.
- Comentarios.
- Líneas en blanco.
- Uso de sangría apropiada.



Conclusión

Un programador debe asociar inmediatamente el proceso de desarrollo de software con el proceso de refinamiento y abstracción que abarca desde el problema real hasta su solución algorítmica con un lenguaje de programación.





COMENCEMOS...

Definición de Lenguaje

El lenguaje es el sistema a través del cual el hombre **comunican sus ideas y sentimientos**, ya sea a través del habla, la escritura u otros signos convencionales
El término lenguaje es de origen latín **lingua**



A que llamamos lenguaje de programación

Un **lenguaje de programación** es un lenguaje formal que proporciona una serie de instrucciones que permiten a un programador escribir secuencias de órdenes y algoritmos a modo de controlar el comportamiento físico y lógico de una computadora con el objetivo de que produzca diversas clases de datos. A todo este conjunto de órdenes y datos escritos mediante un lenguaje de programación se le conoce como programa.

Tipos de lenguajes

Existen dos tipos de lenguajes claramente diferenciados:

Lenguaje de bajo nivel.



Lenguaje de alto nivel.

Tipos de lenguajes

La maquina sólo entiende un lenguaje conocido como código binario o código máquina, consistente en ceros y unos. Es decir, sólo utiliza 0 y 1 para codificar cualquier acción.

Tipos de lenguajes

Los lenguajes más próximos a la arquitectura hardware se denominan lenguajes de bajo nivel y los que se encuentran más cercanos a los programadores y usuarios se denominan lenguajes de alto nivel.

Lenguajes de bajo nivel

Son lenguajes totalmente dependientes de la máquina, es decir que el programa que se realiza con este tipo de lenguajes no se pueden migrar o utilizar en otras máquinas.

Al estar prácticamente diseñados a medida del hardware, aprovechan al máximo las características del mismo.

Dentro de los lenguajes de bajo nivel se encuentra:

El lenguaje maquina: este lenguaje ordena a la máquina las operaciones fundamentales para su funcionamiento. Consiste en la combinación de 0's y 1's para formar las ordenes entendibles por el hardware de la maquina. Este lenguaje es mucho más rápido que los lenguajes de alto nivel. La desventaja es que son bastantes difíciles de manejar.

Dentro de los lenguajes de bajo nivel se encuentra:

El lenguaje ensamblador: es un derivado del lenguaje maquina y esta formado por abreviaturas de letras y números llamadas mnemotécnicos. Con la aparición de este lenguaje se crearon los programas traductores para poder pasar los programas escritos en lenguaje ensamblador a lenguaje máquina. Como ventaja con respecto al código máquina es que los códigos fuentes eran más cortos y los programas creados ocupaban menos memoria, pero seguía siendo difícil de probar y mantener.

Lenguaje de Alto Nivel

Son aquellos que se encuentran más cercanos al lenguaje natural que al lenguaje máquina. Están dirigidos a solucionar problemas mediante el uso de Estructuras Dinámicas de Datos. Se tratan de lenguajes independientes de la arquitectura del ordenador. Por lo que, en principio, un programa escrito en un lenguaje de alto nivel, lo puedes migrar de una máquina a otra sin ningún tipo de problema.

Estos lenguajes permiten al programador olvidarse por completo del funcionamiento interno de la maquina para la que están diseñando el programa.

Tan solo necesitan un traductor que entiendan el código fuente como las características de la maquina.

Podemos llamar HTML como Lenguaje de Programación?

En principio diremos que HTML no es un lenguaje de programación, aunque de forma coloquial muchas veces se escuche referencias a HTML como si lo fuera.

HTML es un lenguaje de etiquetas. Estas etiquetas (tag) HTML comunican al navegador cuál es la información a mostrar por pantalla, además del formato de dicha información.

Ejemplo:

Lenguaje	Código	Salida por pantalla
C	<pre>#include <stdio.h> int main() { int i; for(i=1; i<=10; i++) { printf("%d", i); } return 0; }</pre>	<pre>1 2 3 4 5 6 7 8 9 10</pre>
HTML	<pre><html> <body> <p>1</p> <p>2</p> <p>3</p> <p>4</p> <p>5</p> <p>6</p> <p>7</p> <p>8</p> <p>9</p> <p>10</p> </body> </html></pre>	<pre>1 2 3 4 5 6 7 8 9 10</pre>

Los lenguajes mas usados en la actualidad

Aug 2019	Aug 2018	Change	Programming Language	Ratings
1	1		Java	16.028%
2	2		C	15.154%
3	4	▲	Python	10.020%
4	3	▼	C++	6.057%
5	6	▲	C#	3.842%
6	5	▼	Visual Basic .NET	3.695%
7	8	▲	JavaScript	2.258%
8	7	▼	PHP	2.075%

TIOBE Index for August 2019 (<https://www.tiobe.com/tiobe-index/>)



JAVA

JAVA sigue desde hace muchos años siendo el lenguaje programación más usado, quizás sea por su gran legibilidad y simplicidad.

Actualmente cuenta con más de 9 millones desarrolladores que lo usan y está presente en más de 7 mil millones de dispositivos en todo el mundo.

C

El lenguaje de programación C fue creado entre 1969 y 1972 en los Laboratorios Bell, es uno de los más utilizados en el mundo. Lo utilizan la mayoría de los sistemas operativos lo cual hace que sea un lenguaje muy flexible.

También se utiliza frecuentemente para el desarrollo de aplicaciones de escritorio como por ejemplo GIMP.



C++

C++ es un lenguaje de programación orientado a objetos y una evolución del lenguaje C.

Es un lenguaje muy utilizado para desarrollar programas y paquetes como por ejemplo el paquete de programas de Adobe.





Python

Python es un lenguaje de programación multiplataforma y multiparadigma.

Es muy fácil de utilizar lo cual lo hace un lenguaje de programación ideal para principiantes.



Visual Basic .NET

Visual Basic .NET te da la posibilidad de automatizar sus propios procesos y crear sus propias aplicaciones web.

Es uno de los lenguajes más amigables para los principiantes de programación, sobre todo a comparación de C#.

C#

C# es un lenguaje de programación orientado a objetos fue desarrollado en el año 2000 por Microsoft para ser empleado en una amplia gama de aplicaciones empresariales.

C# es una evolución de los lenguajes de programación C y C++, y destaca por su sencillez.





El lenguaje de programación PHP fue creado en 1994 por el programador canadiense Rasmus Lerdorf.

Es de fácil acceso para nuevos programadores y a su vez ofrece a los más experimentados muchas posibilidades.



Javascript

Javascript es un lenguaje de programación que puede ser utilizado para crear programas que son integrados a una página web o dentro de aplicaciones más grandes.

Además lo podemos utilizar para crear efectos y realizar acciones interactivas.

Algunos ejemplos de este lenguaje son el chat, calculadoras, buscadores de información y un sin fin de utilidades más.

La programación estructurada

La **programación**
estructurada es
un paradigma de
programación orientado a
mejorar la claridad, calidad y
tiempo de desarrollo de
un programa de computadora



La segmentación

En la visión moderna de la programación estructurada, segmentos o módulos que se habrán de integrar.

Una regla practica para lograr este propósito es establecer que cada segmento del programa no exceda, en longitud, de una página de codificación, o sea, alrededor de 50 lineas.

De lo clásico hacia lo práctico

Desarrollar programas usando la técnica llamada refinamientos sucesivos.



El problema inicial es segmentado
Principio de Divide y Conquistarás

La integración debe ser sencilla y no presentar problemas en sí misma.

La programación estructurada propone segregar los procesos en estructuras elementales:

- Secuencia
- Selección
- Iteración