

PROGRAMACIÓN

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

Lic. Mariela A. Velázquez

Etapas en la resolución de problemas con computadora

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

Codificación

- El algoritmo se “traduce” al lenguaje de programación elegido.
- Escribir las instrucciones reales.

Compilación → errores y advertencias.

Corrección, prueba y optimización

Para la ejecución → traducción → lenguaje de máquina

En la compilación y prueba:

Los errores de ejecución.

Los errores de tipo lógico.

Los errores de sintaxis.

Verificación

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.
- Uso de sangría apropiada.





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.

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 máquina para la que están diseñando el programa. Tan solo necesitan un traductor que entienda el código fuente como las características de la máquina.

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) 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

El índice de la comunidad de programación TIOBE es un indicador de la popularidad de los lenguajes de programación. Las calificaciones se basan en la cantidad de informáticos calificados en todo el mundo, cursos y proveedores externos.

Aug 2020	Aug 2019	Change	Programming Language	Ratings	Change
1	2	⬆	C	16.98%	+1.83%
2	1	⬇	Java	14.43%	-1.60%
3	3		Python	9.69%	-0.33%
4	4		C++	6.84%	+0.78%
5	5		C#	4.68%	+0.83%
6	6		Visual Basic	4.66%	+0.97%
7	7		JavaScript	2.87%	+0.62%
8	20	⬆	R	2.79%	+1.97%
9	8	⬇	PHP	2.24%	+0.17%

TIOBE Index for August 2020 (<https://www.tiobe.com/tiobe-index/>)

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.



Ahora les toca trabajar a ustedes

- Busquen rápidamente información sobre los siguientes lenguajes:
JAVA, C++, PYTHON, C#, PHP, JAVASCRIPT



Y lo compartamos
en el siguiente
padlet.

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




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

Lenguaje C

Un poco de historia...

Los laboratorios Bell lo desarrollaron a principios de la década del 70.



Los autores del lenguaje son: Brian Kernighan (Canadá) , y Dennis Ritchie(Estados Unidos).



El objetivo de su creación fue para que los programadores de Bell pudiesen redactar su sistema operativo UNIX para una nueva computadora producida por DEC (Digital Equipment Corporation).

Debido a que los otros lenguajes de alto nivel existentes en aquel tiempo (COBOL, FORTRAN, etc), eran demasiados lentos para ser utilizados en la codificación de un sistema operativo. Los programadores de laboratorios Bell decidieron desarrollar su propio lenguaje, basado en Algol y BCPL, dos eficientes lenguajes de alto nivel.

Características de C

- Es un lenguaje para la programación estructurada.
- Es tipificado.
- Contiene muy pocas palabras reservadas.
- No contiene órdenes para trabajar con objetos compuestos (cadenas, registros, etc).
- Distingue entre mayúsculas y minúsculas.

Ventajas

- Es el lenguaje más portado en existencia, habiendo compiladores para casi todos los sistemas conocidos.
- Proporciona facilidades para realizar programas modulares y/o utilizar código o bibliotecas existentes.
- Es un lenguaje muy flexible, muy veloz y potente, lo que permite un software efectivo.
- Posibilita una programación estructurada o modular.
- Acceso a memoria de bajo nivel mediante el uso de punteros.

Desventajas

- No tiene instrucciones propias para la asignación dinámica de memoria ni instrucciones de entrada/salida. Todas estas operaciones de alto nivel pueden ser realizadas por funciones explícitamente.
- Se requiere más tiempo en conseguir el ejecutable, porque cada vez compila todo el fichero.
- No dispone de sistemas de control automáticos y la seguridad depende casi exclusivamente de la experiencia del programador.

**ENTONCES SATANAS DIJO,
USEMOS EL LENGUAJE C**



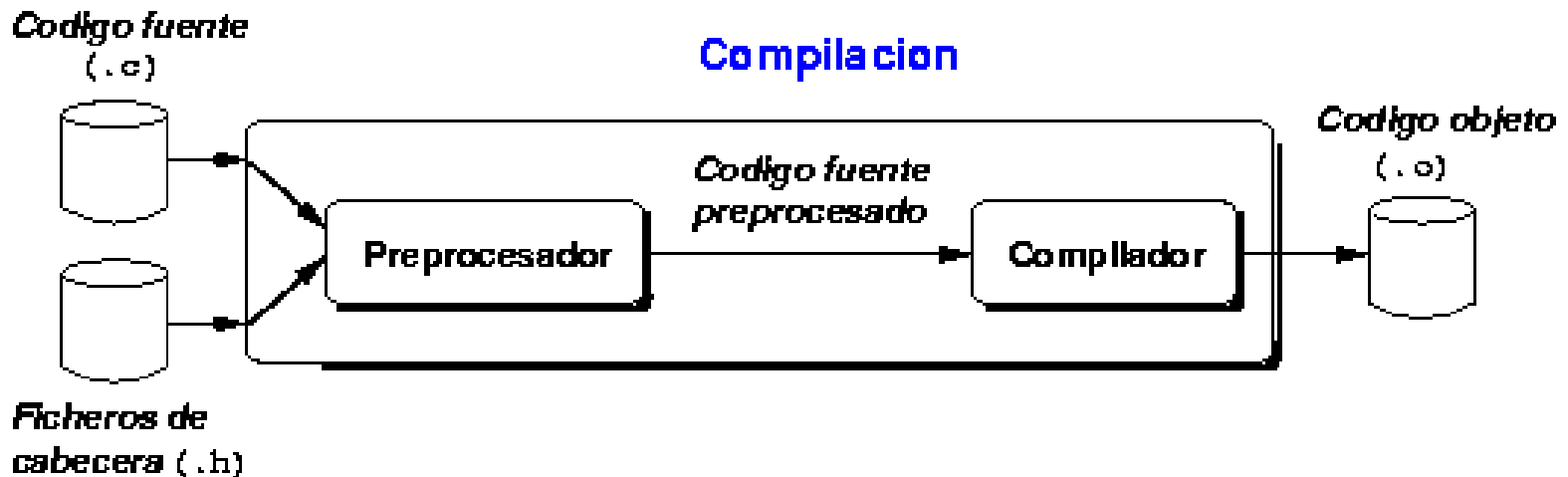
Imagen creada en [GeneradorMemes.com](https://www.generadormemes.com)

En un programa intervienen

- Ordenes para el preprocesador
- Variables
- Constantes
- Aritmética
- Funciones
- Funciones de entrada y salida
- Comentarios

Ordenes para el preprocesador

- Conceptualmente es un paso previo a la compilación.
- Las mas usadas son: # include
define
- Para inclusión de archivos y substitución de macros.



Funciones

“ Un programa escrito en código C es una reunión de funciones “.

main → función principal debe estar , presente en todos los programas escritos en C

main → puede inconvocar a otra funciones .

Función main

- Un método para comunicar datos entre las funciones, es que la función que llama proporciona una lista de valores, llamados argumentos.
- Los paréntesis después del nombre de la función, están para encerrar una lista de valores que serán argumentos.
- Puede ocurrir que una función este definida para ser una función que no espera argumentos, tal es el caso del main, lo cual se indica con una lista de argumentos vacía.

main()

Funciones de Entrada y Salida

La biblioteca estándar `stdio.h` provee al programador de una extensa gama de funciones para lectura y escritura.

Es necesario escribir una orden para el preprocesador para usar dichas funciones.

```
#include <stdio.h>
```

De las variables y constantes

- Objetos sobre los que actúan las instrucciones que componen el programa.
- Deben declararse como tales.
- Deben tener un identificador asociado.
- El 1º carácter debe ser una letra. Letras mayúsculas y minúsculas son diferentes.
- Debe indicarse que tipo de datos pueden contener.
- Las variables pueden inicializarse en forma grupal.
- Las constantes se definen en una línea `#define` , que es una directiva del preprocesador.

#define nombretexto de reemplazo

- Las constantes pueden ser enteras, reales y de carácter.

Aritmética

- Interacción entre los operadores aritméticos y las variables y/o constantes declaradas (si el tipo es numérico).
- El tipo de operación permitido está **ligado tipo de dato** con que fue declarada la variable y/o constante.
- Tipos de datos básicos: int, float, char, short, long, double.

Comentarios

Dos modos de comentar las acciones del código escrito:

// comentarios de una sola línea

/*comentarios de varias líneas

..... */

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Hasta la próxima clase!!\n");
```

```
    return 0;
```

```
}
```