

INTRODUCCIÓN A LA PROGRAMACIÓN

3° P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

LENGUAJES DE PROGRAMACIÓN

Cuando se programa, se indica una serie de instrucciones para que la computadora las ejecute. Para asegurarse que la computadora entienda las instrucciones, se han establecido lenguajes bien definidos para especificar y poder generar la comunicación con la computadora. Estos lenguajes tienen características similares a los lenguajes comunes que utilizan las personas para comunicarse unas con otras, pues cuentan con reglas y estructuras que deben seguirse.

A continuación se expondrá lo que son los lenguajes de computación, el papel que juegan dentro de la programación, herramientas que permiten traducir los algoritmos a instrucciones que puedan ser interpretadas y ejecutadas por las computadoras.

Lenguaje Medio de comunicación entre los seres humanos a través de signos orales y escritos que poseen un significado. También podría decirse que es cualquier procedimiento que sirve para comunicarse, representado mediante símbolos y caracteres específicos

Desde El Punto De Vista Informático El Lenguaje Es: La representación por medio de signos, símbolos y caracteres que existe entre la comunicación de la PC.

¿Qué es un Lenguaje de programación?

En computación, un lenguaje de programación es cualquier lenguaje artificial, el cual, se utiliza para definir adecuadamente una secuencia de instrucciones que puedan ser interpretadas y ejecutadas en una computadora. Se asume que las instrucciones así escritas son traducidas luego a un código que la máquina pueda "comprender". El proceso de traducción es realizado normalmente por la computadora, usando un programa especializado para tal fin.

Los Lenguajes De Programación son los lenguajes utilizados para escribir programas de computadoras que puedan ser entendidos por ellas.

Los lenguajes de programación intentan conservar una similitud con el lenguaje humano, con la finalidad de que sean más naturales a quienes los usan. Establecen un conjunto de reglas sintácticas y semánticas, las cuales rigen la estructura del programa de computación que se escribe o edita. De esta forma, permiten a los programadores o desarrolladores, poder especificar de forma precisa los datos sobre los que se va a actuar, su almacenamiento, transmisión y demás acciones a realizar bajo las distintas circunstancias consideradas.

Los lenguajes de programación se pueden distinguir en tres grandes categorías:

- Máquina
- Bajo nivel (ensamblador) y
- Alto nivel

Uso de los lenguajes de programación:

Una vez que se identifica una tarea y se conoce el algoritmo para resolverla, el programador debe codificarlo en una lista de instrucciones, propias de algún lenguaje de computación. Los programas pueden ser escritos en cualquier lenguaje de la amplia gama disponibles. La escogencia del más adecuado, dependerá de factores como la experiencia del programador con un lenguaje, determinar cuál producirá el programa menos complicado, la flexibilidad y la compatibilidad del programa como resultado del lenguaje usado.

Clasificación de los lenguajes de programación: Niveles y Generaciones:

En computación se suele hablar de niveles y generaciones de los lenguajes de programación. Respecto al nivel, se clasifican en lenguajes de bajo y alto nivel. Los lenguajes de bajo nivel, son aquellos que se asemejan al lenguaje empleado por la computadora, mientras que los lenguajes de alto nivel, son aquellos que se asemejan al lenguaje humano.

INTRODUCCIÓN A LA PROGRAMACIÓN

3° P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

Lenguaje Máquina

Es el lenguaje propio de la computadora, basado en la lógica binaria, de ceros y unos (00010111). Este lenguaje resulta difícil de utilizar para las personas; ya que el programador debe introducir todos y cada uno

de los comandos y datos en forma binaria, y una operación sencilla como comparar el contenido de un registro

con los datos situados en una ubicación del chip de memoria puede tener el siguiente formato: 11001010 00010111 11110101 00101011.

La programación en lenguaje máquina es una tarea tan tediosa y consume tanto tiempo que muy raras veces lo

que se ahorra en la ejecución del programa justifica los días o semanas que se han necesitado para escribir el

mismo.

Niveles de los lenguajes de programación

Los lenguajes de alto nivel son resultado de una evolución, desde los lenguajes de bajo nivel. La computadora sólo es capaz de comprender su lenguaje máquina. Pero para las personas no es fácil entender y manejar este código máquina de combinaciones binarias. Las investigaciones dieron un primer paso que llevó a la evolución desde las hileras de números binarios del código máquina, a un lenguaje ensamblador, que las representaba como una serie de comandos más entendibles. Los lenguajes de nivel superior están orientados más hacia lo que se quiere realizar, que en la manera como se ejecutarán los comandos.

1.- Lenguajes de bajo nivel:

A nivel de circuitos, los microprocesadores sólo son capaces de procesar señales electrónicas binarias. Estas instrucciones binarias que se envían al microprocesador, se componen de series de unos y ceros, espaciadas en el tiempo de una forma determinada. Esta secuencia de señales se denomina código máquina. El código representa normalmente datos y números, así como instrucciones para manipularlos. Para hacer más fácil el manejo del código máquina, se dio a cada instrucción un mnemónico, como por ejemplo STORE (Almacenar), ADD (Adicionar) o JUMP (Saltar). Tal abstracción da como resultado el ensamblador, que es un lenguaje de muy bajo nivel basado en mnemónicos y que es específico para cada microprocesador.

De esta forma, son llamados lenguajes de bajo nivel, a aquellos constituidos por instrucciones en lenguaje máquina o próximos a él. Tales lenguajes permiten crear programas muy rápidos y eficientes, pero en contrapartida, estos lenguajes suelen ser difíciles de aprender, programar y depurar. Un hecho importante es, que los programas escritos en un bajo nivel, son prácticamente específicos para cada procesador, lo que implica, que si se quiere ejecutar el programa en otra máquina de arquitectura diferente, es necesario volver a escribir el programa desde el principio.

Lenguaje De Bajo Nivel (Ensamblador)

Como vimos anteriormente la programación en lenguaje máquina es difícil por ello se necesitan lenguajes que faciliten este proceso. Por este motivo han sido diseñados los lenguajes de bajo nivel. Estos lenguajes dan a cada instrucción un mnemónico, como por ejemplo STORE, ADD o JUMP. Los lenguajes de bajo nivel permiten crear programas muy rápidos, pero que son a menudo difíciles de aprender.

Esta abstracción da como resultado un lenguaje de muy bajo nivel que es específico de cada microprocesador:

El Lenguaje Ensamblador

Al asignar un código mnemotécnico (por lo general de tres letras) a cada comando en lenguaje máquina, es posible escribir y depurar o eliminar los errores lógicos y de datos en los programas escritos en lenguaje ensamblador, empleando para ello sólo una fracción del tiempo necesario para programar en lenguaje máquina.

En el lenguaje ensamblador, cada comando mnemotécnico y sus operadores simbólicos equivalen a una instrucción de máquina. Un programa ensamblador traduce el código fuente, (una lista de códigos de operación mnemotécnicos y de operadores simbólicos), a código objeto (es decir, a lenguaje máquina) y,

INTRODUCCIÓN A LA PROGRAMACIÓN

3° P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

a continuación ejecuta el programa, todo esto gracias a un intérprete o a un compilador, los cuáles veremos más adelante

Sin embargo, el lenguaje ensamblador puede utilizarse con un solo tipo de chip de CPU o microprocesador, por lo que los programas escritos en un bajo nivel son prácticamente específicos para cada procesador.

Si se quiere ejecutar el programa en otra máquina con otra tecnología, será necesario rescribir el programa desde el principio.

Así que los programadores necesitaban un método abreviado en el que un enunciado simbólico pudiera representar una secuencia de numerosas instrucciones en lenguaje máquina, y un método que permitiera que el mismo programa pudiera ejecutarse en varios tipos de máquinas. Estas necesidades llevaron al desarrollo de lenguajes de alto nivel.

2.- Lenguajes de alto nivel:

Al pensar en una computadora, normalmente se concibe como una máquina que realiza tareas de procesamiento de texto y cálculos. Esta es una percepción bastante sintética y esquemática de entender la computadora. Existe un alto nivel de abstracción entre lo que se pide a la computadora y lo que realmente comprende. De un modo paralelo se presenta una relación compleja entre los lenguajes de alto nivel y el código máquina. Un programa escrito en un lenguaje de alto nivel, debe ser compilado o interpretado para traducir su código, en otro de bajo nivel (lenguaje máquina). Como ejemplo de lenguajes de alto nivel se tienen Pascal, BASIC, FORTRAN, COBOL, Java, C y C++.

En los lenguajes de alto nivel las instrucciones son independientes de la máquina, son más fáciles de aprender porque están formados por elementos de lenguajes naturales (del inglés). En BASIC, comandos como "IF CONTADOR = 10 THEN STOP" son usados para ordenar a la computadora, que pare si CONTADOR es igual a 10. Pero aunque las computadoras parecieran comprender el lenguaje natural, lo hacen de una manera bastante rígida y sistemática, lo cual podría parecer un poco frustrante e inspira la búsqueda de lenguajes más cercanos aún al natural.

Estos lenguajes de alto nivel son los que se emplean con mayor frecuencia como lenguajes de programación, porque permiten expresar los algoritmos de una manera y con un estilo fácilmente reconocible por parte de diversos programadores y usuarios; debido a que están formados por elementos de lenguajes naturales, como el inglés utilizando términos del tipo LIST, PRINT u OPEN como comandos.

En Basic, el lenguaje de alto nivel más conocido, los comandos se introducen desde el teclado, desde un programa residente en la memoria o desde un dispositivo de almacenamiento, y son interceptados por un programa que los traduce a instrucciones en lenguaje máquina.

Asimismo, presentan una ventaja fundamental: la facilidad de poder ser transportados de una máquina a otra sin necesidad de realizar grandes cambios en ellos, por lo que se dice que son independientes de la máquina empleada.

A este grupo pertenecen los lenguajes más conocidos, tales como el APL, FORTRAN, PASCAL, COBOL, LISP, PROLOG, C, ADA, PL/I.

Sin embargo, tanto los lenguajes de alto nivel como los de bajo nivel, no son entendibles directamente por la máquina, sino que necesitan ser traducidos a instrucciones en lenguaje máquina que entiendan las computadoras por lo que es necesario disponer de una interfase con el lenguaje máquina para que el programa sea ejecutable. Al respecto existen dos tipos fundamentales de interfase, que son:

- a) Compiladores
- b) Intérpretes

GENERACIONES DE LOS LENGUAJES DE PROGRAMACIÓN

El lenguaje máquina consiste de códigos binarios

▲ **Primera generación:** Lenguaje máquina

Son los lenguajes más básicos. Los lenguajes de máquina se representan en bytes, los cuales, están compuestos de bits, que son números binarios o que consisten únicamente de ceros (0) o unos (1).

INTRODUCCIÓN A LA PROGRAMACIÓN

3º P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

Estos valores pueden corresponder a que la electricidad esté encendida o apagada en la máquina, o a la presencia o ausencia de carga magnética en un medio de almacenamiento. A partir de estos dos estados se forman los esquemas de codificación (como el código ASCII) que permiten generar letras, números, signos de puntuación y caracteres especiales.

Una computadora sólo es capaz de comprender su lenguaje máquina original, el conjunto de instrucciones para realizar sus operaciones elementales.

▲ **Segunda generación:** lenguajes ensambladores:

Están constituidos por nemotécnicos similares a palabras en idioma inglés. En primer lugar, se crea un archivo fuente con las instrucciones que se desea que la computadora ejecute, usando un editor, que es una especie de procesador de palabras. El código fuente es traducido al lenguaje máquina mediante programas traductores (compiladores). Los programadores rara vez escriben programas de tamaño relevante en este tipo de lenguajes, debido a que, a pesar que es mucho más fácil que utilizar código de máquina, siguen siendo altamente detallados (instrucciones muy básicas). Sólo se hace en los casos donde la velocidad es clave (como en la programación de juegos de video) y para afinar partes importantes de los programas que son escritos en lenguajes superiores.

▲ **Tercera generación:** Lenguajes de alto nivel:

Son llamados de alto nivel porque se asemejan más al lenguaje que utilizan los humanos al comunicarse. Fueron desarrollados con la finalidad de facilitar el proceso de programación. Cuentan con comandos que se acercan más a las palabras de uso común en lugar de tener que usar combinaciones binarias del código máquina o los comandos básicos del ensamblador. Estos lenguajes hacen más fácil la lectura, escritura y comprensión de los programas, aunque de manera semejante a los lenguajes ensambladores, deben ser convertidos a lenguaje máquina, para poder ser usados en la computadora.

Los programas de la tercera generación son capaces de soportar programación estructurada, es decir, permiten el uso de estructuras de programación específicas para ramificaciones y ciclos en el flujo del programa. Por otro lado, resultan más claros puesto que usan palabras o comandos semejantes al lenguaje normal, lo que permite compartir el desarrollo entre diferentes programadores. Los programadores que conforman un equipo pueden leer el código realizado por los demás y comprender la lógica y el flujo de control del programa.

Los lenguajes de tercera generación crean programas más claros y estructurados

Entre los lenguajes de tercera generación se incluyen:

- FORTRAN (Traductor de Fórmulas: Formula TRANslator), diseñado específicamente para programas matemáticos y de ingeniería, motivo por el cual, no fue de amplio uso en computadoras personales. Su uso se dirige especialmente a equipos de cómputo dedicados a la investigación y la educación. La versión más actual (1990) fue denominada FORTRAN-90.

- COBOL (Lenguaje simbólico de programación, orientado hacia aplicaciones comerciales y de gestión: Common Business Oriented Language) desarrollado en 1960, con el fin de solucionar las incompatibilidades de los fabricantes de computadoras. Ganó amplia aceptación como lenguaje estandarizado llegando a utilizarse ampliamente en los negocios, pero perdió seguidores en la pasada década. Motivado al problema del año 2000 fueron requeridos programadores especializados en este lenguaje para reprogramar las millones de líneas de código escritas en COBOL para que tales programas continuasen funcionando.
- BASIC (Código de instrucción simbólica universal para principiantes: Beginners Allpurpose Symbolic Instruction Code) comienza como una herramienta para enseñar sobre programación a

INTRODUCCIÓN A LA PROGRAMACIÓN

3º P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

estudiantes. Fue desarrollado por Jhon Kemeny y Thomas Kurtz en 1964. Por su simplicidad se volvió ampliamente popular y usado en educación y entre los programadores aficionados, pero no ha llegado a ser un lenguaje viable para aplicaciones comerciales. Esto último se debe a que no posee un amplio repertorio de herramientas como el que ofrecen otros lenguajes. Además, sus compiladores no producen archivos ejecutables tan compactos, rápidos y eficientes como los que producen otros lenguajes.

- PASCAL, nombrado en honor del inventor francés Blas Pascal, es introducido por Niklaus Wirth en 1971. El Pascal, buscaba superar las limitaciones de los otros lenguajes de programación y demostrar la manera adecuada de implementar un lenguaje de cómputo. Ha llegado a considerarse un excelente lenguaje de enseñanza. Las personas que se inician en la programación le consideran fácil para implementar algoritmos, su compilador es estricto con el uso de programación estructurada, asegurándose de que los errores sean señalados al principio.
- C, desarrollado a principios de los sesenta por Brian Kernighan y Dennis Ritchie, quienes buscaban un mejor lenguaje de integrar con el sistema operativo UNIX, para que los usuarios pudiesen hacer modificaciones y mejoras con facilidad. El código ejecutable obtenido del lenguaje C, es rápido y eficiente. El C es poderoso, puesto que brinda la libertad de conseguir que se realice casi cualquier cosa en una computadora. Esta es una de las razones por la cual se ha vuelto tan popular entre quienes desarrollan software para aplicaciones comerciales profesionalmente. Su dificultad estriba, en que no es tan sencillo de aprender en comparación con otros lenguajes.
- C++, Es un lenguaje de principios de los ochenta desarrollado por Bjarne Stroustrup. Es extremadamente poderoso y eficiente igual que el C. Su aprendizaje implica aprender todo acerca de C y además aprender sobre la programación orientada a objetos y su aplicación mediante el C++. Desde su origen ha ido ganando terreno entre quienes desarrollaban en C.
- Java (1991) es desarrollado por Sun Microsistema. Cuando Internet comenzó a ganar popularidad como red de comunicaciones a mediados de los noventa, Sun redirige a Java para convertirlo en un ambiente de programación que permitiese crear sistemas interactivos y dinámicos (denominados applets) para las páginas Web. La complejidad de Java es similar a C++, pero ha ido ganando popularidad por la demanda de aplicaciones para páginas Web, incluso se estima que puede llegar a desplazar a C++ como ambiente de programación.

• **Cuarta generación:** Lenguajes de propósito especial:

Estos lenguajes de cuarta generación (4GL) son usados en programación, de propósitos específicos, caracterizados por una mayor facilidad de uso comparados con los de la tercera generación, permitiendo la creación de prototipos de una aplicación rápidamente. Los prototipos permiten tener una idea del aspecto y funcionamiento de la aplicación antes que el código sea terminado. Esto implica que quienes estén involucrados en el desarrollo de la aplicación pueden aportar retroalimentación en aspectos como, estructura y diseño desde el principio del proceso. A cambio de esta capacidad para trabajar más velozmente, debe sacrificarse parte de la flexibilidad con la que se disponía con los lenguajes anteriores.

Muchos de estos lenguajes tienen capacidad para bases de datos, permitiendo crear programas que sirvan de enlace con las mismas. Los programas incluyen formas y cuadros para introducir datos, así como solicitar reportes de información de las bases de datos. En general, ahorran tiempo puesto que el código requerido para "conectar" los cuadros de diálogo y las formas se genera de forma automática.

Entre los programas de cuarta generación se tienen:

- ▲ Los programas de ambientes visuales: Existen versiones de programas populares como el Visual BASIC o el Visual C, los cuales, soportan características y métodos orientados a objetos y que permiten crear programas en un ambiente visual. Por ejemplo, si se desea ubicar un cuadro de

INTRODUCCIÓN A LA PROGRAMACIÓN

3º P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

texto en un formulario para entrada de datos sólo se requiere arrastrarlo desde la barra de herramientas hasta la posición deseada sobre el formulario. Con los programas de tercera generación se requiere escribir el código, indicando el tamaño y ubicación exacta del cuadro de texto.

La creación de formularios es más fácil en ambientes visuales.



En el ambiente visual, simplemente se ubica el cuadro en el lugar que se requiere y se da el tamaño adecuado arrastrando sus bordes con el ratón, el código se genera automáticamente. Esto facilita el desarrollo de la interfaz de usuario, acelerando y facilitando la creación de programas.

- ▲ Ambientes de autoría: Conformados por herramientas de programación para propósitos especiales, se usan para crear multimedia, capacitación basada en computadora, páginas Web y otros. Un ambiente de autoría es "Macromedia Director" el cual se usa para crear títulos multimedia que combinen, cortos de música, texto, animación, gráficos y demás. Son ambientes de desarrollo visuales en el cual gran parte del código se escribe automáticamente, pero la mayoría de ambientes de autoría robustos también incluyen sus propios lenguajes, lenguajes de escritura, que proporcionan un control extra sobre el producto final.

- **Quinta generación:** Lenguajes naturales:

Esta generación incluye la inteligencia artificial y sistemas expertos. En lugar de sólo ejecutar un conjunto de órdenes, el objetivo de estos sistemas es "pensar" y anticipar las necesidades de los usuarios. Estos sistemas se encuentran aún en desarrollo. Aunque los sistemas de inteligencia artificial han demostrado ser más complicados para desarrollar de lo que se anticipaba, los investigadores afirman que pronto serán capaces de tomar como entrada hechos y luego usar un procesamiento de datos que formule una respuesta adecuada, de modo similar a como responden los humanos.

Tabla 1.- Las cinco generaciones de lenguajes de programación

Generación	Nombre	Particularidad
Primera	De máquina	Específico para cada microprocesador, uso de código binario
Segunda	Ensamblador	Uso de nemotécnicos que abstraen del lenguaje máquina.
Tercera	De procedimientos	Lenguajes estructurados con comandos cercanos al lenguaje común.
Cuarta	Orientados a procesos	Programas orientados a problemas específicos.
Quinta	Natural	Incluye inteligencia artificial y sistemas expertos.

INTRODUCCIÓN A LA PROGRAMACIÓN

3º P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

La disponibilidad de programas de calidad desarrollados más rápidamente, se ve acrecentada con el pasar del tiempo motivada por dos aspectos. Por un lado, la acumulación del conocimiento de cómo hacer las cosas, organizado mediante técnicas y metodologías bien definidas y estructuradas, que plantean alternativas como guías a seguir. Por otro lado, la disponibilidad de lenguajes de programación los cuales proveen mejores medios para que los desarrolladores sean capaces de aplicar su creatividad, ingenio y experiencia.

Una tendencia marcada desde hace algunos años, es la dependencia creciente de componentes de software reusable por parte de programadores y equipos de desarrollo. El enfoque de la programación orientada a objetos es muestra de ello, así como su incorporación en los lenguajes de programación. De tal modo, es muy probable que incluso lleguen a existir proveedores de objetos componentes de software, que los ofrezcan a los programadores, como en la actualidad se ofrecen componentes hardware. Entonces el enfoque del trabajo de los desarrolladores de software cambiará, tal vez, dividiéndolos en dos grupos.

En la programación cada vez, es más común el uso de componentes reusables

Por un lado, quienes se encarguen de la integración de los objetos cuando se trata de desarrollar programas específicos, más que de desarrollar a detalle cada componente desde el principio. En tanto que los surtidores de tales componentes se especializan en desarrollar los objetos que les son demandados, que podrán ser componentes estándar o a la medida (con especificaciones únicas).

Otra gran influencia en el desarrollo de programas, es resultado de la expansión en el uso de Internet en lo personal y en lo comercial. Es posible que pronto se generalice el uso de programas los cuales se ejecuten en Internet y los desarrolladores produzcan programas que además de enviar y recibir mensajes y archivos, hagan uso de los inmensos recursos mundiales disponibles. Los programadores podrían estar desarrollando programas que se ejecuten en Internet, requiriendo entonces, lenguajes de programación más adaptados a este tipo de empresas.

1. ACTIVIDAD

1) _COMPLETAR SEGÚN CONVenga, CON LA PALABRA ADECUADA.

El lenguaje de programación permite definir una secuencia de		para ser ejecutadas por la computadora.
Los lenguajes de programación, intentan mantener similitud con los lenguajes		.
Los lenguajes de cómputo, establecen reglas		y semánticas para escribir el código.
Los lenguajes de programación, se pueden categorizar en		y generaciones.
Cuando los lenguajes se asemejan más al lenguaje empleado por las computadoras, se habla de lenguajes de nivel		
Se dice que los lenguajes son		a partir de la generación.

INTRODUCCIÓN A LA PROGRAMACIÓN

3° P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

de nivel		
La secuencia de señales binarias que recibe el microprocesador se denomina código		
Un programa escrito en un lenguaje de alto nivel, debe ser		o interpretado para traducir su código a uno de bajo nivel.
Un programa		crea un archivo de código objeto a partir del código fuente.
Un programa		recorre línea a línea el código fuente en el momento de ejecución.

 **BASADO EN LO QUE SE HA APRENDIDO, SELECCIONE LA OPCIÓN MÁS ACERTADA EN CADA CASO MARCANDO CON UNA CRUZ AL FINAL DE LA MISMA:**

1- Los lenguajes de máquina y ensambladores se encuentran:

- En la primera generación
- En la segunda generación
- En generaciones diferentes
- Entre los lenguajes de alto nivel

2- Los ensambladores usan como instrucciones

- Ceros y unos
- Código y objeto
- Compiladores
- Mnemotécnicos

3-Para escribir el archivo fuente se utiliza:

- Un editor
- Un compilador
- Señales binarias
- Un enlazador

4-El uso del ensamblador en la actualidad, aunque puede ser complicado, suele estar motivado por la siguiente situación:

- Cuando la velocidad del programa es clave
- Cuando se debe crear código objeto
- Cuando se debe trabajar en lenguajes de alto nivel
- Cuando se requiere facilidad de uso

5-Los lenguajes de tercera generación fueron creados para :

- Las aplicaciones Internet
- Promover el uso del código de máquina
- Facilitar el proceso de programación
- Tener una alternativa al lenguaje máquina

6-Los programas que comienzan a ser capaces de soportar programación estructurada, son los lenguajes de:

- Primera generación
- Tercera generación
- Cuarta generación

INTRODUCCIÓN A LA PROGRAMACIÓN

3° P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

Ninguna de las anteriores

7- El lenguaje de programación FORTRAN tiene como característica:

Popularidad en las áreas de la ciencia y la ingeniería

Facilidad para los principiantes

Orientado a las hacia las operaciones comerciales

No ser viable para a educación

8- El lenguaje considerado como excelente para la enseñanza al demostrar la manera adecuada de implementar un programa, se trata de:

BASIC

UNIX

PASCAL

C

9- El C++ se caracteriza dentro de los lenguajes de tercera generación por:

Incorporar la programación orientada a objetos

Ser una aplicación atractiva y muy fácil de usar

Capacidad para aplicaciones de páginas Web

Un lenguaje enfocado a la gestión comercial

10- Cuando se habla de programación para propósitos específicos, debemos pensar en:

Cobol

Lenguajes de bajo nivel

Lenguajes de 4ta generación

Lenguajes de 3ra generación

INTRODUCCIÓN A LA PROGRAMACIÓN

3º P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

COMPILADORES - INTÉRPRETES - ENSAMBLADORES

Se ha señalado que la computadora sólo puede ejecutar instrucciones en código de máquina, por lo cual, todos los programas escritos en cualquiera de los lenguajes de programación deben ser convertidos o traducidos. Este proceso es llevado a cabo por unos programas especiales, los "compiladores" y los "intérpretes". Para darse una idea de cómo funcionan, es útil comparar su actuación con la de un profesional humano encargado de traducir. Un compilador equivale a que el traductor humano, partiendo de un texto, prepare otro texto nuevo e independiente, traducido a la otra lengua. En contraste, un intérprete equivale a que tal profesional traduzca de forma hablada, cada grupo de palabras o instrucción, a medida que la oye, sin dejarlas por escrito.

La traducción del código fuente, escrito en lenguaje ensamblador, a código máquina (o código objeto), no representa un proceso complicado y se lleva a cabo mediante un compilador. De manera similar, la traducción de código fuente de lenguajes de alto nivel puede ser realizada por un compilador, o también mediante un intérprete. Con base en el código fuente suministrado, el compilador crea una lista de instrucciones de código máquina:

Un traductor que facilita la comunicación entre el programador y la máquina, por medio de un proceso de transformación llamado compilación.

De esta manera traduce un programa íntegro a lenguaje máquina antes de su ejecución, por lo cual se ejecutan con tanta rapidez como si hubiesen sido escritos directamente en lenguaje máquina.

El compilador es el más eficaz para la mayor parte de las máquinas, puesto que presenta la ventaja de que cada una de las sentencias del programa es interpretada y traducida al lenguaje máquina solo una vez.

Un compilador crea una lista de instrucciones de código máquina, el código objeto, basándose en un código fuente.

El código objeto resultante es un programa rápido y listo para funcionar, pero que puede hacer que falle el ordenador si no está bien diseñado.

Proceso de compilación

El código objeto. Este código objeto obtenido, es un programa rápido y ya listo para funcionar, pero está expuesto a poseer errores que pueden provocar fallas en la computadora. En el proceso de compilación pueden resultar varios archivos de código objeto, los cuales deben enlazarse para obtener un archivo ejecutable. Esto se logra mediante un programa llamado enlazador.

Por otra parte, los intérpretes recorren línea a línea el código fuente en el momento de su ejecución. Esto los hace más lentos, debido a la necesidad de traducir el programa mientras se ejecuta, pero a cambio, son más flexibles como entornos de programación y depuración. Es natural que los lenguajes de programación utilicen sólo uno de estos métodos para producir el código fuente, sin embargo muchos también disponen de versiones interpretadas y compiladas.

En cuanto a las generaciones de lenguaje, se tienen cinco diferentes, considerando que cada generación sucesiva se establece con lenguajes más poderosos y fáciles de usar que los de la generación precedente. Los lenguajes calificados como de alto nivel, comienzan a partir de la tercera generación.

Los compiladores, los intérpretes y los ensambladores se encargan de traducir lo que haya escrito en lenguaje de alto nivel (código fuente) y lo convierten a código objeto (casi ejecutable).



INTRODUCCIÓN A LA PROGRAMACIÓN

3º P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

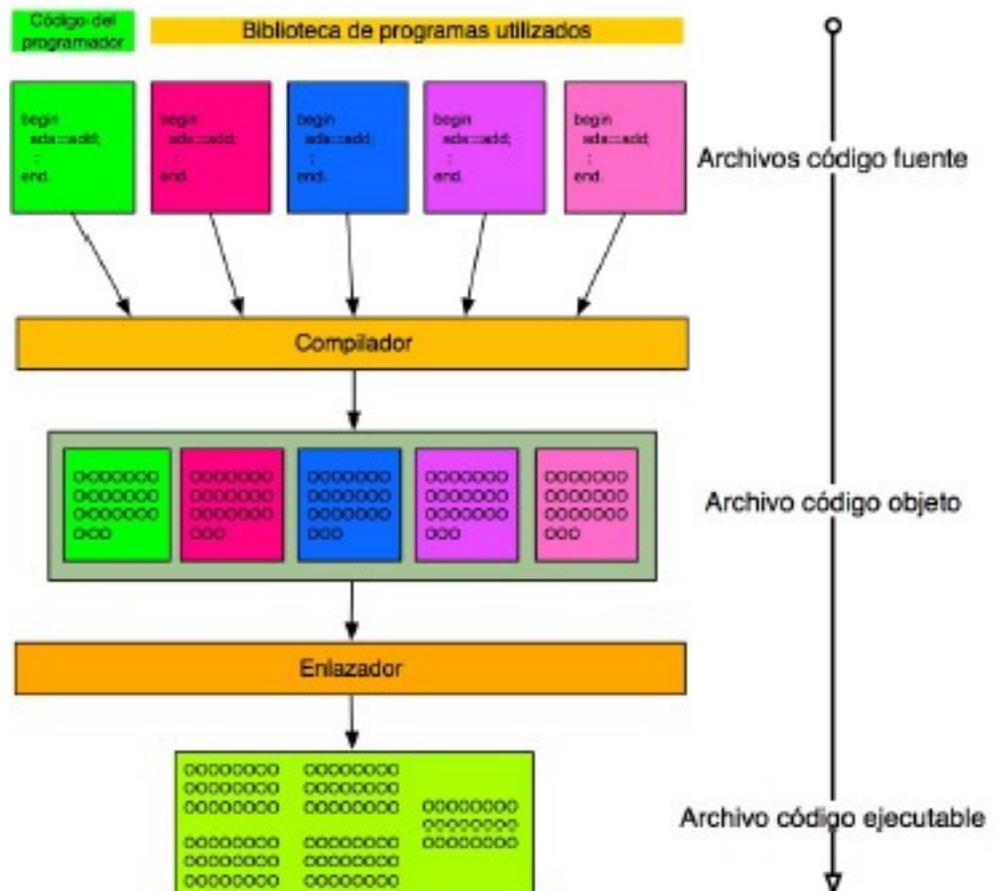
Docente Barrenechea, María

COMPILADOR

Es un programa que traduce un programa escrito en un lenguaje de alto nivel, por ejemplo C++, en un programa en lenguaje de máquina que la computadora es capaz de entender y ejecutar directamente. Un compilador es un tipo especial de programa, en cuanto a que sus entradas o datos son algún programa y su salida es otro programa.

Para evitar confusiones, solemos llamar programa fuente o código fuente al programa de entrada, y programa objeto o código objeto a la versión traducida que el compilador produce.

Código se usa frecuentemente para referirse a un programa o a una parte de él, sobre todo cuando se habla de programas objeto. Ejemplo:

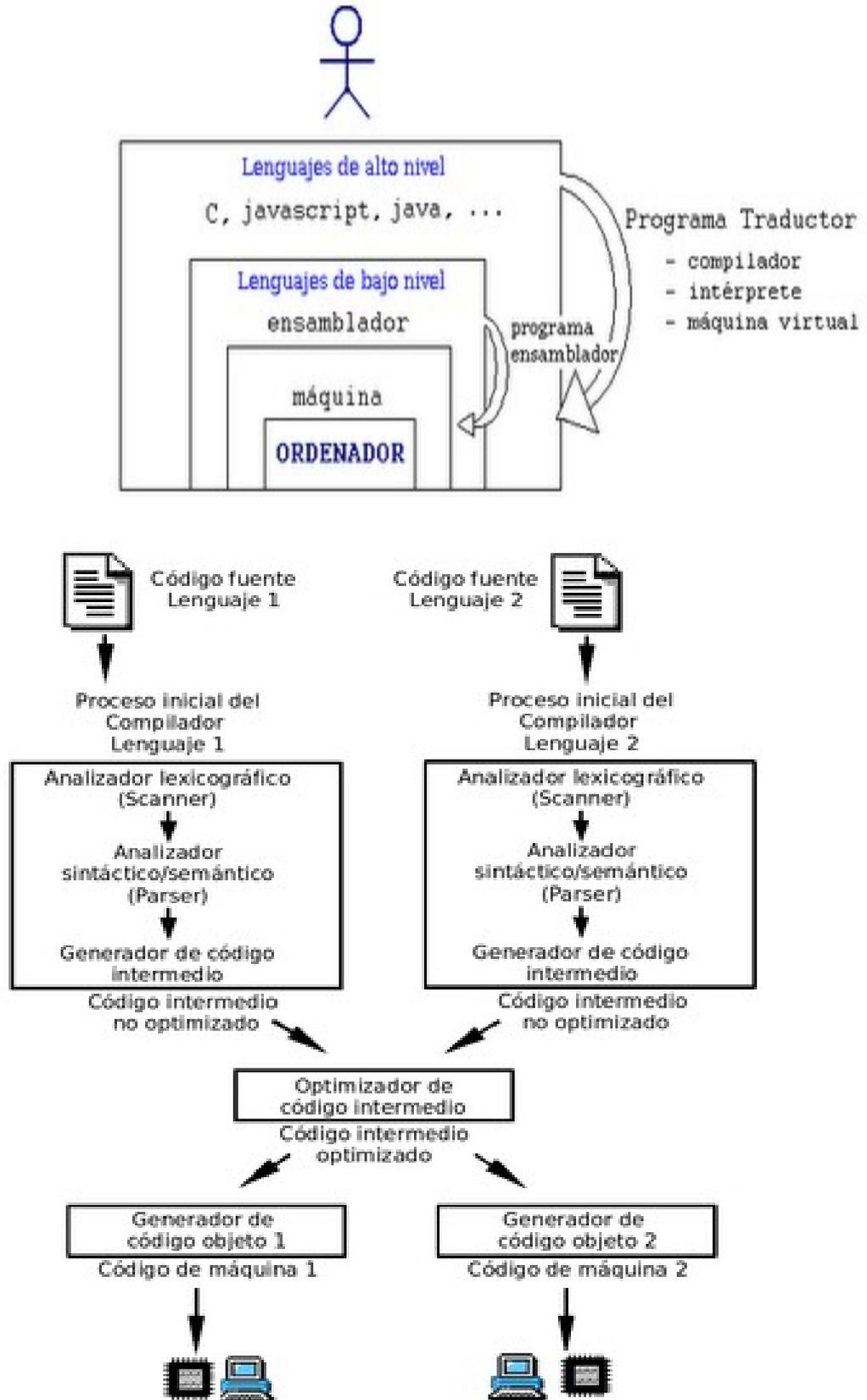


INTRODUCCIÓN A LA PROGRAMACIÓN

3º P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

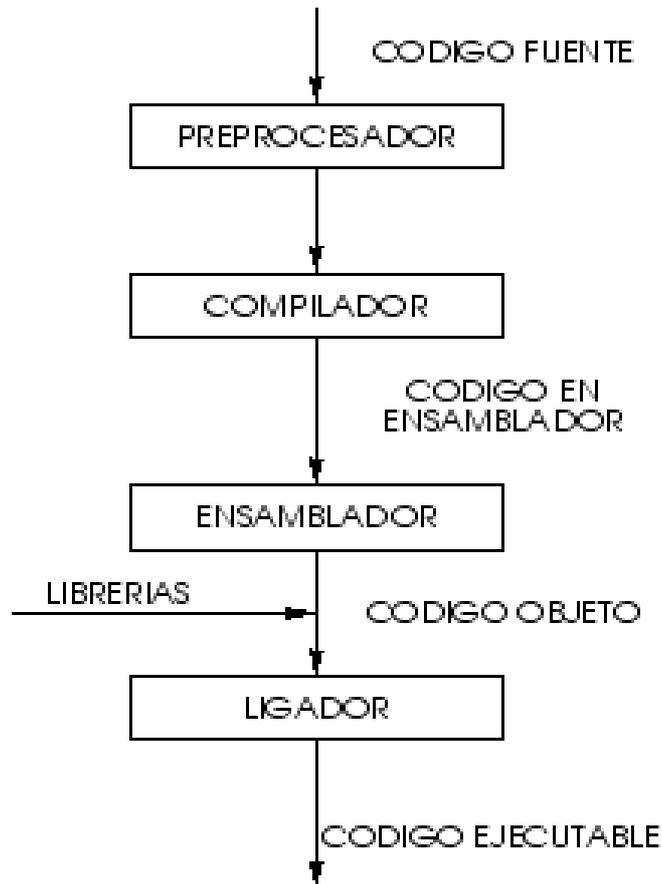


INTRODUCCIÓN A LA PROGRAMACIÓN

3° P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María



El compilador, informa al usuario de la presencia de errores en el programa fuente, pasándose a crear el programa objeto cuando está libre de errores.

El código objeto puede ser ejecutado posteriormente.

Una vez traducido un programa, su ejecución es independiente de su compilación. involucra dos pasos en su operación:

- ☞ Convertir código fuente a objeto
- ☞ Ejecutar el código objeto

Ventaja:

Al tener el código objeto, el programa se ejecuta más rápido

FASES DE COMPILACIÓN

Análisis: Dependiente del lenguaje. Independiente de la máquina

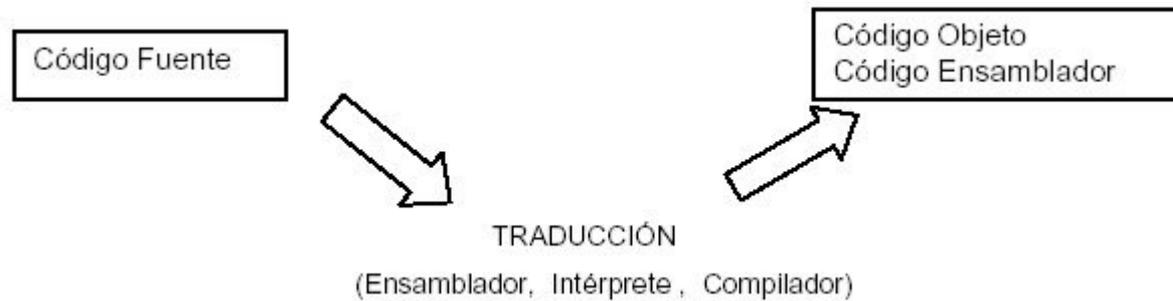
INTRODUCCIÓN A LA PROGRAMACIÓN

3º P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

Sintaxis: Independiente del lenguaje. Dependiente de la máquina.



INTÉRPRETE:

Es un programa que se traduce línea por línea bajo la misma plataforma.

Es un traductor pero más lento que los compiladores ya que no producen un código objeto, sino que recorren el código fuente una línea cada vez. Cada línea se traduce a código máquina y se ejecuta. Cuando la línea se lee por segunda vez, como en el caso de los programas en que se reutilizan partes del código, debe compilarse.

Aunque este proceso es más lento, es menos susceptible de provocar fallos en la computadora.

Es el que permite que un programa fuente escrito en un lenguaje vaya traduciéndose y ejecutándose directamente sentencia a sentencia por la computadora. Convierte uno por uno los enunciados del código fuente a código objeto antes de ser ejecutados. Convierte y ejecuta el programa en línea al mismo tiempo. Ejemplo: Basic estándar.

Ventaja:

Las ventajas de los intérpretes son:

- ▲ Resulta más fácil localizar y corregir errores (depuración de programas)
- ▲ son más pedagógicos para aprender a programar.
- ▲ El programa es más fácil de desarrollar.
- ▲ Traducen programas de alto nivel. No se genera en la mayoría de los ficheros.

Para cada una de las líneas se ejecuta el siguiente proceso:

1. Análisis de la instrucción de esa línea
2. Traducción de esa línea (si ya está correcta) a código objeto
3. Ejecución de esa línea

Con el intérprete, cada vez que necesitamos ejecutar el programa tenemos que volver a analizarlo porque no hay código objeto.

Con el compilador, aunque más lenta la traducción, sólo se realiza una vez.

Ejemplo 4

Supongamos que se han almacenado tres valores $5 = 01012$, $25 = 110012$ y $58 = 1110102$ en las posiciones de memoria con direcciones 4, 5 y 6.

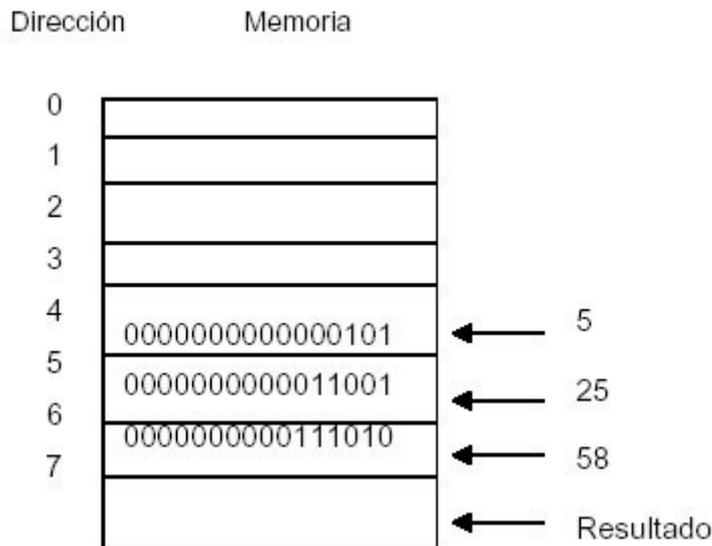
Queremos multiplicar los dos primeros valores, sumar el tercero y almacenar el resultado en la palabra de memoria 7.

INTRODUCCIÓN A LA PROGRAMACIÓN

3° P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María



Para llevar a cabo este cálculo, se deben ejecutar las siguientes instrucciones:

Recuperar el contenido de la palabra de memoria 4 y cargarlo en el registro acumulador de la unidad aritmético lógica.

Recuperar el contenido de la palabra de memoria 5 y calcular el producto de este valor y el valor situado en el acumulador.

Recuperar el contenido de la palabra de memoria 6 y sumar su valor con el valor situado en el registro acumulador.

Almacenar el contenido del registro acumulador en la palabra de memoria 7. Para almacenar estas instrucciones en la memoria de la computadora, deben estar representadas en forma binaria. Las direcciones de los datos no presentan problemas, puesto que pueden ser convertidos fácilmente a direcciones binarias:

4 = 100

5 = 101

6 = 110

7 = 111

Las operaciones de cargar, multiplicar, sumar, almacenar y otras instrucciones máquina básicas se representan mediante códigos numéricos, llamados códigos de operación, por ejemplo:

LOAD = 16 = 10000

STORE = 17 = 10001

ADD = 35 = 100011

MULTIPLY = 36 = 100100

SUB = 37 = 100101

DIV = 38 = 100110

Usando parte de una palabra para almacenar el código de operación y otra para la dirección del operando, podemos representar nuestra secuencia de instrucciones en lenguaje máquina como sigue:

1. 000100000000100

2. 001001000000101

3. 001000110000110

4. 000100010000111

Estas pueden ser almacenadas en cuatro palabras consecutivas de memoria. Cuando se ejecuta el programa, la unidad de control recuperará cada una de las instrucciones, la decodificará para determinar la operación y la dirección del operando, recuperará el operando, y entonces ejecutará la operación requerida, usando la unidad aritmético lógica cuando sea necesario. Los programas para las primeras computadoras tuvieron que ser escritos en lenguaje de máquina. Posteriormente fue posible escribirlos

INTRODUCCIÓN A LA PROGRAMACIÓN

3º P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

en lenguaje ensamblador usando códigos nemotécnicos en lugar de códigos de operación numéricos y nombres de variables en lugar de direcciones numéricas.

Por ejemplo la secuencia de instrucciones anteriores se escribiría así:

1. LOAD A
2. MULT B
3. ADD C
4. STORE X

Luego que se crearon los lenguajes de alto nivel, las instrucciones se escribían en forma más entendible para el programador. El ejemplo anterior podría ser como lo siguiente usando C++:

$X = A * B + C$

El producto en la programación se representa por asterisco. En cada uno de estos casos (ensamblador y lenguajes de alto nivel), el compilador traduce cada instrucción del programa en una secuencia de cuatro instrucciones máquina y genera un programa objeto.

Ejercicios Propuestos:

Usando mnemónicos de instrucción y códigos de operación, escriba una secuencia de instrucciones en a) Lenguaje ensamblador b) Lenguaje de máquina, equivalente a las instrucciones de C++:

1. $X = (A - B) * C$
 $X = (A+B) / (C+D)$
 $X = (A + B) - C$

Para las instrucciones máquina, suponga que los valores de A, B, C y D son almacenados en las palabras de memoria 15,16, 17 y 18 respectivamente, y que los valores de X e Y se almacenarán en la palabra de memoria 23 y 24 respectivamente

1. ACTIVIDAD

- Busca en Internet videos educativos sobre el tema: Compiladores, Intérpretes, Ensambladores. Coloca la dirección URL de dichos videos.
- A continuación coloca una breve descripción de lo que muestra y enseña el video.

Clasificaciones según complejidad

Las estructuras algorítmicas son las herramientas que con las cuales se entregan las instrucciones más precisas al computador.

Las estructuras algorítmicas se dividen en tres categorías, secuenciales, de decisión y cíclicas.

Las **estructuras secuenciales** se utilizan en problemas simples, donde solo se necesita seguir una serie de pasos en forma ordenada para llegar al resultado que se desea.

Por otro lado las **estructuras de decisión**, se utilizan para darle el poder de decisión a un algoritmo o un cierto nivel de inteligencia, es en base a estas estructuras que un algoritmo puede tomar ciertas decisiones y realizar un análisis más exhaustivo de los datos.

Además existen las **estructuras cíclicas**, las cuales se utilizan para ejecutar un bloque de código por un número limitado de ocasiones.

Clasificación general de las estructuras algorítmicas.



INTRODUCCIÓN A LA PROGRAMACIÓN

3° P.B.yS. INFORMÁTICA

Complejo Educativo N° 394 "Dr. F. de Gurruchaga"
Eugenia

Docente Barrenechea, María

EJEMPLO ESTRUCTURA SECUENCIAL: Y CONDICIONAL

https://ambientes.inacap.cl/Cursos/SCORMAICC/C925114654/html/4_4.asp

CLASIFICACIÓN POR COMPLEJIDAD DE LAS ESTRUCTURAS

https://ambientes.inacap.cl/Cursos/SCORMAICC/C925114654/html/5_1.asp

ESTRUCTURAS SECUENCIALES

https://ambientes.inacap.cl/Cursos/SCORMAICC/C925114654/html/5_2.asp

ESTRUCTURAS CONDICIONALES

https://ambientes.inacap.cl/Cursos/SCORMAICC/C925114654/html/5_3.asp

ESTRUCTURAS CICLICAS

https://ambientes.inacap.cl/Cursos/SCORMAICC/C925114654/html/5_4.asp