



Lenguajes de programación

Entre los lenguajes de programación más conocidos se encuentran:

Lenguajes de máquina:

Son el nivel más básico de los lenguajes de programación. Fueron la primera generación dentro de esta categoría.

Desventajas de los lenguajes de máquina:

- Los programas tenían que escribirse usando códigos binarios únicos para cada computadora.
- Los programadores debían tener un conocimiento detallado de las operaciones internas del tipo específico de CPU que estaban utilizando.
- La programación era difícil y propensa a errores
- Los programas no son portátiles a otras computadoras.

Lenguajes del ensamblador:

Son la segunda generación de lenguajes de máquina. Fueron desarrollados para reducir las dificultades de escribir programas en lenguaje de máquina. El lenguaje ensamblador también es un lenguaje de bajo nivel (se refiere a los recursos de la máquina, como registros y direcciones de memoria), también es específico de un modelo de computadora o una serie de modelos. Un programa en lenguaje ensamblador es traducido a un lenguaje de máquina por un simple traductor llamado ensamblador. Los lenguajes ensambladores se usan hoy en día solo cuando se requiere un control estricto sobre los recursos de hardware de la computadora, como en ciertos programas de sistemas, particularmente aquellos para computación en tiempo real.

_Ventajas del Ensamblador.

- Utiliza instrucciones codificadas simbólicas que son más fáciles de recordar.
- La programación se simplifica ya que un programador no necesita conocer la ubicación exacta de almacenamiento de datos e instrucciones.
- El uso eficiente de los recursos informáticos se ve contrarrestado por los altos costos del desarrollo de sistemas muy tediosos y por el bloqueo de la portabilidad del programa.

Desventaja del Ensamblador

- Los lenguajes ensambladores son únicos para tipos específicos de computadoras.
- Los programas no son portátiles a otras computadoras.

Lenguajes de alto nivel (procedimiento)

son los lenguajes de programación de tercera generación. Estos lenguajes proporcionan declaraciones, cada una de las cuales se traduce a varias instrucciones en lenguaje de máquina. Los lenguajes de alto nivel incluyen COBOL (programas de aplicaciones comerciales), BASIC (usuarios finales de microcomputadoras), FORTRAN (aplicaciones científicas y de ingeniería), y los más populares hoy en día son C, C++ y Visual Basic.

Ventajas

- Más fácil de aprender y comprender que un lenguaje ensamblador como instrucciones (declaraciones) que se asemejan al lenguaje humano o la notación estándar de las matemáticas.
- Tener reglas, formas y sintaxis menos rígidas, por lo que se reduce la posibilidad de error.
- Son programas independientes de la máquina, por lo que los programas escritos en un lenguaje de alto nivel no tienen que ser reprogramados cuando se instala una nueva computadora.
- Los programadores no tienen que aprender un nuevo lenguaje para cada computadora que programan.

Desventajas

- Menos eficientes que los programas en lenguaje ensamblador y requieren una mayor cantidad de tiempo de computadora para la traducción a instrucciones de máquina.

Más allá de los lenguajes de programación de alto nivel:

Los lenguajes de cuarta generación (4GL) especifican lo que se debe hacer en lugar de detallar los pasos para hacerlo. Los 4GL incluyen una variedad de lenguajes de programación que son menos procedimentales y conversacionales que los lenguajes anteriores.

Ventajas

- Simplificó el proceso de programación.
- Usar lenguajes no procedimentales que alienten a los usuarios y programadores a especificar los resultados que desean, mientras que las computadoras determinan la secuencia de instrucciones que lograrán esos resultados.
- Utilizar lenguajes naturales que no impongan reglas gramaticales rígidas.

Desventajas

- Menos flexible que otros idiomas
- Menos eficiente (en términos de velocidad de procesamiento y cantidad de capacidad de almacenamiento necesaria).

Varias lenguas podrían pretender pertenecer a la quinta generación. Es probable que los siguientes tipos de lenguajes de programación influyan en el desarrollo de este nuevo paradigma:

- Los lenguajes de programación orientada a objetos (POO) vinculan elementos de datos y los procedimientos o acciones que se realizan en ellos, en objetos. Los ejemplos incluyen Smalltalk, C++, Visual Basic, Java, Turbo C++, C++, Object C+

- Lenguajes que facilitan el procesamiento paralelo en sistemas con gran cantidad de procesadores.
- Lenguajes funcionales (como LISP), basados en el concepto matemático de computación como aplicación de funciones.
- Subconjuntos limitados de lenguajes naturales que pueden ser procesados gracias al avance de la inteligencia artificial.

Ventajas

- Los lenguajes OOP son más fáciles de usar y más eficientes para programar la interfaz de usuario orientada a gráficos requerida por muchas aplicaciones.
- Los objetos programados son reutilizables.

Traductores: compiladores e intérpretes

Hay disponible una variedad de paquetes de software para ayudar a los programadores a desarrollar programas de computadora. Por ejemplo, los traductores de lenguajes de programación son programas que traducen otros programas a códigos de instrucción de lenguaje de máquina que las computadoras pueden ejecutar. Otros paquetes de software llamados herramientas de programación ayudan a los programadores a escribir programas proporcionando facilidades de creación y edición de programas. Los programas de traducción de idiomas (procesadores de idiomas) son programas que traducen otros programas a códigos de instrucción de lenguaje de máquina que la computadora puede ejecutar. Estos programas le permiten escribir sus propios programas proporcionando funciones de creación y edición de programas. Los programas de traducción de lenguajes de programación son conocidos por una variedad de nombres.

- **Ensamblador:** traduce los códigos de instrucciones simbólicas de los programas escritos en un lenguaje ensamblador a instrucciones en lenguaje máquina.
- **Compilador:** traduce (compila) declaraciones de lenguaje de alto nivel (programas fuente) a programas de lenguaje de máquina
- **Intérprete:** traduce y ejecuta cada declaración del programa una a la vez, en lugar de producir primero un programa completo en lenguaje de máquina, como lo hacen los compiladores y ensambladores.

Lenguajes de Cuarta Generación: (4GL's)

Los 4GL incluyen una variedad de lenguajes de programación que son menos procedimentales y conversacionales que los lenguajes anteriores. El uso de lenguajes de cuarta generación permite multiplicar por varias la productividad en el desarrollo de sistemas de información.

Categorías de lenguajes de cuarta generación y su función en la informática del usuario final

La característica distintiva de los 4GL es que especifican lo que se debe hacer en lugar de como hacerlo. Las características de los 4GL incluyen:

- Los lenguajes son no procedimentales.
- No especifican el procedimiento completo para realizar la tarea (esto lo completa el traductor de software para el 4GL).

- Se requiere alrededor de una décima parte del número de instrucciones en 4GL en comparación con los lenguajes de procedimiento.
- Las categorías principales de 4GL son lenguajes de consulta, generadores de informes y generadores de aplicaciones.
- Los lenguajes de consulta y los generadores de informes hacen que sea innecesario desarrollar ciertas aplicaciones al proporcionar acceso directo a una base de datos. Los generadores de aplicaciones hacen que sea relativamente fácil especificar en términos no procedimentales un sistema para dicho acceso.
- Los 4GL también tienen generadores de software para producir sistemas de apoyo a la toma de decisiones e información ejecutiva.

Tres categorías de 4GL son:

- Idioma de consulta.
- Generadores de informes.
- Generadores de aplicaciones.

Idiomas de consulta: Los lenguajes de consulta permiten a los usuarios finales acceder a las bases de datos directamente. Las características del lenguaje de consulta incluyen:

- Utilizado en línea para consultas ad-hoc, es decir, consultas que no están predefinidas
- El resultado de la consulta generalmente no tiene formato, ya que se muestra en un formato predeterminado seleccionado por el propio sistema.
- La interacción suele ser simple, solo se involucran cálculos muy simples.
- La mayoría de los lenguajes de consulta también permiten actualizar las bases de datos. Con muchos lenguajes de consulta, es posible solicitar una salida gráfica a una consulta.

Seis estilos principales para consultar una base de datos:

- Relleno del formulario.
- Selección de menú.
- Lenguaje de consulta de tipo comando, como SQL.
- Consulta por ejemplo (QBE).
- Manipulación directa.
- Lenguaje natural restringido.

Los lenguajes de consulta son apropiados para:

- • Cajeros automáticos.
- • Kioscos electrónicos.

Generadores de informes: Un generador de informes permite que un usuario final o un profesional de sistemas de información produzca un informe sin detallar todos los pasos necesarios, como formatear el documento. Las características de los generadores de informes incluyen:

- Ofrezca a los usuarios un mayor control sobre el contenido y la apariencia de la salida que un lenguaje de consulta.
- Los datos especificados pueden recuperarse de los archivos o bases de datos especificados, agruparse, ordenarse y resumirse de una manera específica y formatearse para imprimirse como se desee.

Generadores de aplicaciones: Un generador de aplicaciones permite especificar una aplicación completa, que consta de varios programas, sin una codificación muy detallada. Las características de los generadores de aplicaciones incluyen:

- La mayoría de los generadores producen (generan) código en un lenguaje procedimental. Este código se puede modificar para satisfacer las necesidades precisas de la aplicación.
- Los generadores dirigidos a los usuarios finales son fáciles de usar. Están dirigidos a un dominio de aplicación limitado. Producen el código principalmente a partir de una especificación de la estructura de archivos y bases de datos y de los diseños dados de pantallas e informes. El procesamiento requerido se especifica en términos naturales para los usuarios finales.
- Una función de pintura de pantalla hace posible especificar la interfaz gráfica de usuario para el sistema en desarrollo.
- Los potentes generadores de aplicaciones requieren la experiencia de los profesionales de sistemas de información y son herramientas de propósito general. A menudo pueden ejecutarse en mainframes y minicomputadoras.

Los generadores de aplicaciones se integran cada vez más en entornos de ingeniería de software asistida por computadora (CASE).

Ventajas

- Simplificó el proceso de programación.
- Usar lenguajes no procedimentales que alienten a los usuarios y programadores a especificar los resultados que desean, mientras que las computadoras determinan la secuencia de instrucciones que lograrán esos resultados.
- Utilizar lenguajes naturales que no impongan reglas gramaticales rígidas.

Desventajas

- Menos flexible que otros idiomas
- Los programas escritos en 4GL generalmente son mucho menos eficientes durante la ejecución del programa que los programas en lenguajes de alto nivel. Por lo tanto, su uso se limita a proyectos que no requieren tal eficiencia.

Lenguajes orientados a objetos:

La idea de la programación orientada a objetos (POO) es construir programas de objetos de software, con el fin de vincular elementos de datos y los procedimientos o acciones que se realizarán en ellos, en objetos. Los ejemplos incluyen Smalltalk, C++, Turbo C++, Object C+, Java.

Las características de la programación orientada a objetos incluyen:

- OOP, los objetos combinan (encapsulan) los datos con las operaciones que actúan sobre los datos.
- Definición y herencia de clases de soporte de programación orientada a objetos, creación de objetos como instancias de clases, envío de mensajes a los métodos en estos objetos durante la ejecución del programa y otras características de programación orientada a objetos.
- OOP simplifica el diseño de sistemas multimedia y de interfaces gráficas de usuario.

Tres conceptos fundamentales de la programación orientada a objetos son:

- **Objetos:** son los componentes básicos de los que se construyen los programas. En software, un componente de programa que modela un objeto del mundo real encapsulando datos e instrucciones que funcionan con estos datos.
- **Clase:** es una plantilla a partir de la cual se crean objetos. Las clases se pueden definir en una jerarquía.
- **Herencia:** en la programación orientada a objetos, las clases inferiores en la jerarquía heredan propiedades (atributos y métodos) de las clases superiores.

Ventajas

- Los lenguajes OOP son más fáciles de usar y más eficientes para programar la interfaz de usuario orientada a gráficos requerida por muchas aplicaciones.
- Ahorra mucha programación, ya que heredar propiedades significa que los objetos programados son reutilizables.