C#

El estándar ECMA-334 lista las siguientes metas en el diseño para C#:

* Lenguaje de programación orientado a objetos simple, moderno y de propósito general.
* Inclusión de principios de [ingeniería de software](https://es.wikipedia.org/wiki/Ingenier%C3%ADa_de_software) tales como revisión estricta de los tipos de datos, revisión de límites de vectores, detección de intentos de usar variables no inicializadas, y recolección de basura automática.
* Capacidad para desarrollar [componentes de software](https://es.wikipedia.org/wiki/Componente_de_software) que se puedan usar en ambientes distribuidos.
* Portabilidad del código fuente.
* Fácil migración del programador al nuevo lenguaje, especialmente para programadores familiarizados con C, C++ y Java.
* Soporte para [internacionalización](https://es.wikipedia.org/wiki/Internacionalizaci%C3%B3n_y_localizaci%C3%B3n).
* Adecuación para escribir aplicaciones de cualquier tamaño: desde las más grandes y sofisticadas como [sistemas operativos](https://es.wikipedia.org/wiki/Sistema_operativo) hasta las más pequeñas funciones.
* Aplicaciones económicas en cuanto a memoria y procesado.

C++

La intención de su creación fue el extender al [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [C](https://es.wikipedia.org/wiki/C_(lenguaje_de_programaci%C3%B3n)) mecanismos que permiten la manipulación de[objetos](https://es.wikipedia.org/wiki/Objeto_(programaci%C3%B3n)). En ese sentido, desde el punto de vista de los [lenguajes orientados a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), el C++ es un lenguaje híbrido.

Posteriormente se añadieron facilidades de [programación genérica](https://es.wikipedia.org/wiki/Programaci%C3%B3n_gen%C3%A9rica), que se sumaron a los paradigmas de[programación estructurada](https://es.wikipedia.org/wiki/Programaci%C3%B3n_estructurada) y [programación orientada a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos). Por esto se suele decir que el C++ es un [lenguaje de programación multiparadigma](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_multiparadigma).

Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Existen también algunos intérpretes, tales como ROOT.

Una particularidad del C++ es la posibilidad de redefinir los [operadores](https://es.wikipedia.org/wiki/Operadores_de_C_y_C%2B%2B), y de poder crear nuevos [tipos](https://es.wikipedia.org/wiki/Tipo_de_datos) que se comporten como tipos fundamentales.

El nombre C++ fue propuesto por [Rick Mascitti](https://es.wikipedia.org/w/index.php?title=Rick_Mascitti&action=edit&redlink=1) en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases". En C++, la expresión "C++" significa "incremento de C" y se refiere a que C++ es una extensión de C.

Binario

Se usa para movimiento de datos y operaciones nada mas

Python

Python es un [lenguaje de programación interpretado](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_interpretado) cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

Se trata de un lenguaje de programación [multiparadigma](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_multiparadigma" \o "Lenguaje de programación multiparadigma), ya que soporta [orientación a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos), [programación imperativa](https://es.wikipedia.org/wiki/Programaci%C3%B3n_imperativa) y, en menor medida, [programación funcional](https://es.wikipedia.org/wiki/Programaci%C3%B3n_funcional). Es un [lenguaje interpretado](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_interpretado), usa [tipado dinámico](https://es.wikipedia.org/wiki/Tipado_din%C3%A1mico" \o "Tipado dinámico) y es[multiplataforma](https://es.wikipedia.org/wiki/Multiplataforma).

Filosofía

* Bello es mejor que feo.
* Explícito es mejor que implícito.
* Simple es mejor que complejo.
* Complejo es mejor que complicado.
* Plano es mejor que anidado.
* Disperso es mejor que denso.
* La legibilidad cuenta.
* Los casos especiales no son tan especiales como para quebrantar las reglas.
* Lo práctico gana a lo puro.
* Los errores nunca deberían dejarse pasar silenciosamente.
* A menos que hayan sido silenciados explícitamente.
* Frente a la ambigüedad, rechaza la tentación de adivinar.
* Debería haber una -y preferiblemente sólo una- manera obvia de hacerlo.
* Aunque esa manera puede no ser obvia al principio a menos que usted sea holandés.[15](https://es.wikipedia.org/wiki/Python#cite_note-15)
* Ahora es mejor que nunca.
* Aunque nunca es a menudo mejor que ya mismo.
* Si la implementación es difícil de explicar, es una mala idea.
* Si la implementación es fácil de explicar, puede que sea una buena idea.

JavaScript

Se utiliza principalmente en su forma del [lado del cliente (client-side)](https://es.wikipedia.org/wiki/Lado_del_cliente), implementado como parte de un [navegador web](https://es.wikipedia.org/wiki/Navegador_web) permitiendo mejoras en la [interfaz de usuario](https://es.wikipedia.org/wiki/Interfaz_de_usuario) y [páginas web](https://es.wikipedia.org/wiki/P%C3%A1gina_web) dinámicas[4](https://es.wikipedia.org/wiki/JavaScript#cite_note-4) aunque existe una forma de JavaScript del [lado del servidor](https://es.wikipedia.org/wiki/Script_del_lado_del_servidor) ([Server-side JavaScript](https://es.wikipedia.org/w/index.php?title=Server-side_JavaScript&action=edit&redlink=1) o [SSJS](https://es.wikipedia.org/w/index.php?title=SSJS&action=edit&redlink=1)). Su uso en [aplicaciones](https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_inform%C3%A1tica) externas a la [web](https://es.wikipedia.org/wiki/World_Wide_Web), por ejemplo en documentos [PDF](https://es.wikipedia.org/wiki/PDF), aplicaciones de escritorio (mayoritariamente [widgets](https://es.wikipedia.org/wiki/Widget" \o "Widget)) es también significativo.

JavaScript se diseñó con una sintaxis similar al [C](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_C), aunque adopta nombres y convenciones del [lenguaje de programación Java](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java). Sin embargo [Java](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java) y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del [Document Object Model](https://es.wikipedia.org/wiki/Document_Object_Model" \o "Document Object Model) (DOM).

Tradicionalmente se venía utilizando en páginas web [HTML](https://es.wikipedia.org/wiki/HTML) para realizar operaciones y únicamente en el marco de la [aplicación cliente](https://es.wikipedia.org/wiki/Cliente_(inform%C3%A1tica)), sin acceso a funciones del [servidor](https://es.wikipedia.org/wiki/Servidor). Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como [AJAX](https://es.wikipedia.org/wiki/AJAX). JavaScript se interpreta en el[agente de usuario](https://es.wikipedia.org/wiki/Agentes_de_usuario) al mismo tiempo que las sentencias van descargándose junto con el código [HTML](https://es.wikipedia.org/wiki/HTML).

Java

Java es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) de [propósito general](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_de_prop%C3%B3sito_general), [concurrente](https://es.wikipedia.org/wiki/Computaci%C3%B3n_concurrente), [orientado a objetos](https://es.wikipedia.org/wiki/Programaci%C3%B3n_orientada_a_objetos) que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los [desarrolladores](https://es.wikipedia.org/wiki/Desarrollador_de_software) de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere"), lo que quiere decir que el [código](https://es.wikipedia.org/wiki/C%C3%B3digo_fuente) que es ejecutado en una plataforma no tiene que ser [recompilado](https://es.wikipedia.org/wiki/Compilaci%C3%B3n_en_tiempo_de_ejecuci%C3%B3n) para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de [cliente-servidor](https://es.wikipedia.org/wiki/Cliente-servidor) de web, con unos 10 millones de usuarios reportados.[1](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)#cite_note-1) [2](https://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)#cite_note-2)

El lenguaje de programación Java fue originalmente desarrollado por [James Gosling](https://es.wikipedia.org/wiki/James_Gosling) de [Sun Microsystems](https://es.wikipedia.org/wiki/Sun_Microsystems" \o "Sun Microsystems) (la cual fue adquirida por la compañía [Oracle](https://es.wikipedia.org/wiki/Oracle_Corporation)) y publicado en 1995 como un componente fundamental de la [plataforma Java](https://es.wikipedia.org/wiki/Java_(plataforma_de_software)) de Sun Microsystems. Su [sintaxis](https://es.wikipedia.org/w/index.php?title=Sintaxis_(programaci%C3%B3n)&action=edit&redlink=1) deriva en gran medida de [C](https://es.wikipedia.org/wiki/C_(programming_language)) y [C++](https://es.wikipedia.org/wiki/C%2B%2B), pero tiene menos utilidades de [bajo nivel](https://es.wikipedia.org/wiki/Lenguaje_de_bajo_nivel) que cualquiera de ellos. Las aplicaciones de Java son generalmente [compiladas](https://es.wikipedia.org/wiki/Compilador) a [bytecode](https://es.wikipedia.org/wiki/Bytecode_Java" \o "Bytecode Java) ([clase Java](https://es.wikipedia.org/w/index.php?title=Clase_Java&action=edit&redlink=1)) que puede ejecutarse en cualquier [máquina virtual Java](https://es.wikipedia.org/wiki/M%C3%A1quina_virtual_Java) (JVM) sin importar la [arquitectura de la computadora](https://es.wikipedia.org/wiki/Arquitectura_de_computadoras) subyacente.

## 

## Críticas

### General

### Java no ha aportado capacidades estándares para aritmética en punto flotante. El estándar [IEEE 754](https://es.wikipedia.org/wiki/IEEE_754) para “Estándar para Aritmética Binaria en Punto Flotante” apareció en 1985, y desde entonces es el estándar para la industria. Y aunque la aritmética flotante de Java (cosa que cambió desde el 13 de noviembre de 2006, cuando se abrió el código fuente y se adoptó la licencia [GPL](https://es.wikipedia.org/wiki/GPL), aparte de la ya existente)[[cita requerida](https://es.wikipedia.org/wiki/Wikipedia:Verificabilidad)] se basa en gran medida en la norma del IEEE, no soporta aún algunas características. Más información al respecto puede encontrarse en la sección final de enlaces externos.

### El lenguaje

### En un sentido estricto, Java no es un lenguaje absolutamente orientado a objetos, a diferencia de, por ejemplo, [Ruby](https://es.wikipedia.org/wiki/Ruby) o [Smalltalk](https://es.wikipedia.org/wiki/Smalltalk" \o "Smalltalk). Por motivos de eficiencia, Java ha relajado en cierta medida el paradigma de orientación a objetos, y así por ejemplo, no todos los valores son objetos.

### El código Java puede ser a veces redundante en comparación con otros lenguajes. Esto es en parte debido a las frecuentes declaraciones de tipos y conversiones de tipo manual (casting). También se debe a que no se dispone de operadores sobrecargados, y a una sintaxis relativamente simple. Sin embargo, J2SE 5.0 introduce elementos para tratar de reducir la redundancia, como una nueva construcción para los bucles ‘’’foreach’’’.

### A diferencia de C++, Java no dispone de operadores de sobrecarga definidos por el usuario. Los diseñadores de Java tomaron esta decisión puesto que consideraban que, bajo ciertas circunstancias, esta característica podía complicar la lectura y mantenimiento de los programas.

### Apariencia

* La apariencia externa (el ‘‘‘look and feel’’’) de las aplicaciones GUI (Graphical User Interface) escritas en Java usando la plataforma Swing difiere a menudo de la que muestran aplicaciones nativas. Aunque el programador puede usar el juego de herramientas AWT (Abstract Windowing Toolkit) que genera objetos gráficos de la plataforma nativa, el AWT no es capaz de funciones gráficas avanzadas sin sacrificar la portabilidad entre plataformas; ya que cada una tiene un conjunto de APIs distinto, especialmente para objetos gráficos de alto nivel. Las herramientas de Swing, escritas completamente en Java, evitan este problema construyendo los objetos gráficos a partir de los mecanismos de dibujo básicos que deben estar disponibles en todas las plataformas. El inconveniente es el trabajo extra requerido para conseguir la misma apariencia de la plataforma destino. Aunque esto es posible (usando GTK+ y el Look-and-Feel de Windows), la mayoría de los usuarios no saben cómo cambiar la apariencia que se proporciona por defecto por aquella que se adapta a la de la plataforma.

### Rendimiento

* El bytecode de Java puede ser interpretado en tiempo de ejecución por la máquina virtual, o bien compilado al cargarse el programa, o durante la propia ejecución, para generar código nativo que se ejecuta directamente sobre el hardware. Si es interpretado, será más lento que usando el código máquina intrínseco de la plataforma destino. Si es compilado, durante la carga inicial o la ejecución, la penalización está en el tiempo necesario para llevar a cabo la compilación.
* Algunas características del propio lenguaje conllevan una penalización en tiempo, aunque no son únicas de Java. Algunas de ellas son el chequeo de los límites de arrays, chequeo en tiempo de ejecución de tipos, y la indirección de [funciones virtuales](https://es.wikipedia.org/wiki/Funci%C3%B3n_virtual).
* El uso de un recolector de basura para eliminar de forma automática aquellos objetos no requeridos, añade una sobrecarga que puede afectar al rendimiento, o ser apenas apreciable, dependiendo de la tecnología del recolector y de la aplicación en concreto. Las JVM modernas usan recolectores de basura que gracias a rápidos algoritmos de manejo de memoria, consiguen que algunas aplicaciones puedan ejecutarse más eficientemente.
* El rendimiento entre un compilador JIT y los compiladores nativos puede ser parecido, aunque la distinción no está clara en este punto. La compilación mediante el JIT puede consumir un tiempo apreciable, un inconveniente principalmente para aplicaciones de corta duración o con gran cantidad de código. Sin embargo, una vez compilado, el rendimiento del programa puede ser comparable al que consiguen compiladores nativos de la plataforma destino, inclusive en tareas numéricas. Aunque Java no permite la expansión manual de llamadas a métodos, muchos compiladores JIT realizan esta optimización durante la carga de la aplicación y pueden aprovechar información del entorno en tiempo de ejecución para llevar a cabo transformaciones eficientes durante la propia ejecución de la aplicación. Esta recompilación dinámica, como la que proporciona la máquina virtual HotSpot de Sun, puede llegar a mejorar el resultado de compiladores estáticos tradicionales, gracias a los datos que sólo están disponibles durante el tiempo de ejecución.
* Java fue diseñado para ofrecer seguridad y portabilidad, y no ofrece acceso directo al hardware de la arquitectura ni al espacio de direcciones. Java no soporta expansión de [código ensamblador](https://es.wikipedia.org/wiki/C%C3%B3digo_ensamblador), aunque las aplicaciones pueden acceder a características de bajo nivel usando bibliotecas nativas (JNI, Java Native Interfaces).

Ruby

Características

* [Orientado a objetos](https://es.wikipedia.org/wiki/Orientado_a_objetos)
* Cuatro niveles de ámbito de [variable](https://es.wikipedia.org/wiki/Variable_(programaci%C3%B3n)): global, clase, instancia y local.
* [Manejo de excepciones](https://es.wikipedia.org/wiki/Manejo_de_excepciones)
* [iteradores](https://es.wikipedia.org/wiki/Iterador_(patr%C3%B3n_de_dise%C3%B1o)) y [clausuras o closures](https://es.wikipedia.org/wiki/Clausura_(inform%C3%A1tica)) (pasando bloques de código)
* [expresiones regulares](https://es.wikipedia.org/wiki/Expresi%C3%B3n_regular) nativas similares a las de [Perl](https://es.wikipedia.org/wiki/Perl) a nivel del lenguaje
* Posibilidad de redefinir los operadores (sobrecarga de operadores)
* [recolección de basura automática](https://es.wikipedia.org/wiki/Recolector_de_basura)
* Altamente [portable](https://es.wikipedia.org/wiki/Portable)
* Hilos de ejecución simultáneos en todas las plataformas usando green threads
* Carga dinámica de [DLL](https://es.wikipedia.org/wiki/Biblioteca_de_enlace_din%C3%A1mico)/[bibliotecas compartidas](https://es.wikipedia.org/wiki/Biblioteca_(programaci%C3%B3n)) en la mayoría de las plataformas
* Introspección, [reflexión](https://es.wikipedia.org/wiki/Reflexi%C3%B3n_(inform%C3%A1tica)) y [metaprogramación](https://es.wikipedia.org/wiki/Metaprogramaci%C3%B3n" \o "Metaprogramación)
* Amplia librería estándar
* Soporta [inyección de dependencias](https://es.wikipedia.org/wiki/Inyecci%C3%B3n_de_dependencias)
* Soporta alteración de objetos en tiempo de ejecución
* [continuaciones](https://es.wikipedia.org/w/index.php?title=Continuaci%C3%B3n&action=edit&redlink=1) y [generadores](https://es.wikipedia.org/w/index.php?title=Generadores&action=edit&redlink=1)