

C++

C++, los recursos del sistema no se adquieren y/o liberan ‘a espaldas’ del programador, sino que éste tiene que especificar exactamente cuándo esto debe ocurrir. Es cierto que C++ tiene estructuras que permiten, hasta cierto punto, agilizar el proceso de desarrollo y disminuir las redundancias en el código (por ejemplo, las plantillas o los *smart pointers*), pero esto no cambia el hecho de que dichas estructuras (y la adquisición de recursos) sigan reglas muy definidas que el programador debe conocer. Todo esto, aunque podría representar una inconveniencia en ciertos contextos, permite que C++ pueda prescindir de un entorno de ejecución y compilarse a instrucciones que son directamente ejecutables por el procesador.

Todo programador sabe que C++ es difícil. Sí, es gigantesco y tiene un montón de peculiaridades. Pero ¿qué obtenemos a cambio? Alto rendimiento + control absoluto sobre todos los procesos. Una vez más, no podemos dejar de mencionar el estándar de C++ 11 con su sencilla sintaxis, algoritmos, y demás características útiles diseñadas para hacer que la vida y el trabajo de los programadores sea menos estresante.

A pesar del hecho de que C++ es un lenguaje muy rico y flexible, su ámbito de uso es bastante limitado. Sus competidores contribuyen a esto. Tal vez lo veamos más claro con un ejemplo en la vida real, el desarrollo de software para móviles. Las principales plataformas ya están ocupadas por lenguajes específicos: Windows Phone por C#, Java por Android, iOS por Objective-C. Tal vez hayas pensado que en plataformas móviles no se pueda utilizar C++. Estás equivocado. La pregunta es si es eficiente y productivo programar en C++ para esas plataformas.

Tampoco C++ es muy adecuado para el desarrollo web, el cual se está haciendo más y más popular cada año. Sí, hay medios para trabajar con sockets y librerías como Wt. Pero, ¿has oído hablar mucho de aplicaciones web desarrolladas en C++? Yo tampoco. Teniendo en cuenta que el desarrollo de software para móviles y la web se han convertido en un campo muy popular últimamente, no es de extrañar que los programadores novatos prefieran aprender otros lenguajes.

C#

C# es un lenguaje *administrado*; en otras palabras, aunque facilita muchas cosas al programador, es totalmente dependiente de su entorno de ejecución^[1] para adquirir o liberar recursos del sistema, como la memoria.

Así que, de entrada, no puedes programar en C# para dispositivos que no tengan instalado el entorno de ejecución que mencionado, o en los que a veces es incluso imposible instalarlo, como es el caso de muchos microcontroladores de bajo consumo, que cuentan con especificaciones muy limitadas de almacenamiento, memoria o velocidad del reloj.

Es muy fácil desarrollar en C# para Windows, pero si lo que quieres es hacer aplicaciones multiplataforma (que funcionen tanto en Windows como en Linux, Mac y a veces hasta en entornos móviles), la cosa cambia.

Aunque C# y las especificaciones de su entorno de ejecución son un estándar^[2] (y no 'pertenece' a Microsoft, como podría pensarse), su implementación fuera del entorno de Windows siempre ha sido muy limitada.

Ha habido implementaciones de software libre como Mono y más recientemente .NET Core, pero las herramientas existentes para C# siguen sin ser demasiado adecuadas, o bien están ligadas a una plataforma específica.

Si quieres, por ejemplo, desarrollar en C# una aplicación multiplataforma cuya interfaz vaya más allá de la línea de comandos, con controles gráficos, siento decirte que tus opciones son bastante limitadas. Las únicas opciones factibles que conozco (y vaya que he investigado bastante) son:

- Hacer un sitio web disfrazado de aplicación. Frameworks como Electron. Esto tiene la seria desventaja de que la interfaz gráfica de la aplicación corre sobre un navegador web, lo cual consume muchísima memoria y es potencialmente lento.
- Usar GTK#, que son *bindings* del entorno GTK+ para .NET; Sin embargo, el proyecto me parece prácticamente abandonado...Pero podrías intentarlo.

C

El lenguaje C es usado para crear programas de software para computadoras en empresas, hogar, juegos y otras áreas. Junto con el lenguaje BASIC, el C y sus posteriores derivados predominan en la programación, excepto en las áreas muy especializadas, tales como en las aplicaciones científicas.

El lenguaje C es uno de los lenguajes de programación más ampliamente utilizados en el mundo, ejecutándose en la mayoría de los sistemas operativos y pueden ser usados en casi todas las plataformas informáticas.

El C es más tolerante a los errores de programación que algunos otros lenguajes, lo que significa que la programación descuidada puede causar consecuencias no deseadas.

Java

Java es tanto un lenguaje de programación como una plataforma informática y un amplio abanico de tecnologías. Originalmente perteneció a la empresa Sun Microsystems, para después ser adquirida por la compañía Oracle.

Java, como plataforma informática, es la base de diversos programas y aplicaciones, ya que el lenguaje que se utiliza es compatible con diversas plataformas y sistemas de arquitectura PC y Mac, de 16, 32 o 64 bits.

Java es una tecnología con compatibilidad para sistemas operativos como OS, Unix, Linux, Solaris, Windows o Android. Por ello, varias aplicaciones para dispositivos móviles con estos sistemas operativos se sirven de la tecnología Java para funcionar.

Java es una tecnología utilizada para crear y desarrollar diversos programas y aplicaciones; muchas de las que se encuentran en dispositivos móviles (como los teléfonos inteligentes, tabletas y otros) usan tecnología Java para funcionar.

Esto es más patente en aplicaciones destinadas a la navegación por internet, en donde entra en servicio como plugin, instalable en la computadora u otros dispositivos, para la mejor ejecución de programas y aplicaciones, entre las que se pueden contar:

- Videojuegos
- Servicios de chat
- Diversas páginas online