

Universidad Politécnica De Chiapas

**CARRERA**: Ingeniería en Desarrollo de Software

**GRUPO**: 1° B

**ALUMNO**: Erik Toledo Trinidad

**MATRICULA:** 191228

**MATERIA:** Expresión Oral y Escrita

**TAREA**: Ensayo

**FECHA DE ENTREGA**: Domingo 27 de enero del 2019

**MAESTRO:** Dr.Guillermo Salazar Arias

Lenguaje de programación C++ y sus carencias.

Introducción

Por: Erik Toledo Trinidad

C++ es un [lenguaje de programación](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) diseñado en 1979 por [Bjarne Stroustrup](https://es.wikipedia.org/wiki/Bjarne_Stroustrup). La intención de su creación fue extender al lenguaje de programación [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.

Se trata de un lenguaje de tipos de datos estáticos. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en [ensamblador](https://es.wikipedia.org/wiki/Lenguaje_ensamblador) con código C++ o acceder directamente a [memoria](https://es.wikipedia.org/wiki/Memoria_de_ordenador) o [dispositivos periféricos](https://es.wikipedia.org/wiki/Dispositivos_perif%C3%A9ricos).

No obstante, tiene carencias en los lenguajes de tipo de dato, que, a su vez presentan problemas para la comunidad que desarrolla con este lenguaje. Existen diversos lenguajes de la programación que reemplazan ah este. Java es uno de los candidatos a reemplazarlos por su función completa en el ámbito laboral. C++ es un lenguaje con el que se puede programar, pero no aplica a la parte de orientación de objetos, sin embargo, en la actualidad, la programación orientada a objetos es una realidad, que, si bien un lenguaje lo impide, tal cual es el C++, existen otros dispuestos a poder realizar esa acción, por eso, el lenguaje C++ podría quedar obsoleto en un par de años por todo lo que le falta para ser un lenguaje completo para el desarrollador.

C++ surge de fusionar dos ideas: la eficiencia del lenguaje C++ para poder acceder al hardware al ejecutar tareas que realmente demandaban recursos de memoria; y las ideas que representan las él nuevos conceptos de clases y objetos. Presenta grandes herramientas de desarrollo para los programadores como las funciones, bibliotecas, clases y los objetos. De manera que el programador se ocupa de utilizar dichas herramientas para resolver un problema específico. C++ es igual: Es un lenguaje bastante simple, pero ha logrado sobrevivir el paso del tiempo porque con él se ha logrado escribir sistemas operativos, compiladores, herramientas de sistema y montón de cosas necesarias para que aplicaciones escritas en lenguajes de más alto nivel puedan funcionar.

Desarrollo

C++ es un lenguaje de tipos de datos estáticos, débilmente tipificado, de medio nivel, ya que dispone de las estructuras típicas de los lenguajes de alto nivel, pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Sin embargo, representa un gran reto poder realizar proyectos en él, pues carece de ciertos aspectos que llevan los desarrolladores a abandonarlo. Este se encuentra muy rezagado a la hora de desarrollar, es comúnmente reemplazado por otros lenguajes como: “Java, Java Script, Python, etc.”. La fluidez del compilador, la optimización de la memoria virtual y el poder evitar la recolección de basura nativa es vital cuando se trabaja en algún proyecto, esto alivia la carga de trabajo humano y en muchas ocasiones evita errores, aunque también supone mayor carga de trabajo para el procesador.

La programación alberga demasiados lenguajes, lo que hace que haya competencia entre ellos, y algunos con carencias como el lenguaje C++, empiecen a quedar en el olvido. Cada uno de estos se esfuerza por optimizarse y mostrar caracterizas que los puedan llevar como el mejor lenguaje y así dejar atrás a la competencia.

Java representa todos los lenguajes, como el mejor, tiene todo lo que un desarrollador busca, se habla de un leguaje completo y por todo lo que te ofrece, es reconocido y ocupado mundialmente.

Existen proyectos que están basados aun en el lenguaje C++ que carecen de, por lo tanto, eso representa una oportunidad para poder implementar el lenguaje Java y con ello obtener muchas mejoras. Las universidades que tienen carreras orientadas a la programación, son objetivos claros para poder dar a conocer este lenguaje, con ello sus funciones y características por el cual lo hacen un lenguaje sencillo y de categoría mundial. Por otro lado, existen demasiados lenguajes, como la experiencia que cada uno de ellos ofrecen, son más completos y eficaces en proyectos simples o de mayor envergadura, el cual, al ser más eficaz, ofrece a los desarrolladores un lenguaje de programación innovador.

El lenguaje C++ ah perdido su lugar como lenguaje, por provocar errores en empresas grandes y ser uno de los que menos soporte tiene. Actualmente hay plataformas que, omiten agregar el lenguaje por que esta siendo excluido de la comunidad prefiriendo Java o Java Script. Hay mas oportunidad de trabajo para un desarrollador que domine Java, tiene mayor monetización, así mismo, les abre las puertas a empresas sumamente importantes dentro de la programación. En el mundo de la programación, existen procesos de entrada y salida, este es otro gran problema que surgió en el 2001 con este lenguaje, ocasionaba que el desarrollador no pudiera hacer esos procesos a la hora de la programación, y a la hora de poner en función el proyecto, marcaba error. La implementación del lenguaje Java suplanto de inmediatamente a C++, puesto a que este lenguaje no tenia errores en los procesos, se ejecutaba sin errores, por eso, este lenguaje, Java, esta siendo el mejor hasta su época.

Para hacer trabajar en aplicaciones, no es recomendable trabajar en este lenguaje, al hacer punteros, este es muy complicado al querer guardar los datos, puesto a que su sistema es demasiado confuso. Lenguajes como Swift es el ideal para tratar con este tipo de trabajos, es super fácil y sencillo el poder guardar información en los punteros y no crea ningún error al querer guardar información.

Conclusión

Ante lo ya leído, podemos saber que, el lenguaje C++ esta siendo perdido y siendo de desagrado para la comunidad de desarrolladores, sin embargo, existen mas lenguajes que suplantan a este. Ese lenguaje pasara a ser retirado acabó de unos años, por su poca eficiencia en el ámbito de la programación y desarrollo de software.

Cabe destacar que, a pesar de ser un lenguaje problemático, existen apps y sistemas que aún están basados en el lenguaje, pero que, en un tiempo corto serán reemplazados por los nuevos lenguajes que tienen mayor eficacia. Los desarrolladores que manejan el lenguaje C++, tendrán que poner de su parte y actualizarse, pues ese lenguaje no es muy común ya dentro del ámbito de desarrollo, hoy en día tiene mejor oportunidad de trabajo alguien que maneja Java que C++.

El C++ además de ser un leguaje problemático, también es mucho más difícil de usar que otros lenguajes de programación., para la gran mayoría programar difícil programar bases de datos. Para los punteros (es muy complicado su sistema). Existen gran mayoría de problemas que podría enlistar, trabajar con este lenguaje no es nada recomendable. Así mismo el tener un lenguaje correcto para aplicar a los programas es parte integral de estos, las desventajas de desconocer un lenguaje o no conocer lo que lo distingue de otros lenguajes es un desfavorecimiento para el programador. Por eso se usa otro lenguaje de programación que no sea C++, debido a que permite tener una compilación mucho más rápida y un uso para el usuario nada tedioso. Aunque no se genera un entorno gráfico muy cómodo para el usuario, este solo se encarga de ingresar la información para obtener el informe que necesita por lo cual en realidad no es un punto en contra el del lenguaje.

Siendo sincero, lo considero un **lenguaje de dificultad media**. Yo sinceramente **no lo recomendaría como primer lenguaje para aprender a programar**. Efectivamente, C++ sienta unas bases teóricas que son excelentes para el que quiera dedicarse a esto de la programación; pero también es cierto que su enfoque de “medio nivel”, más próximo a las características internas de las computadoras, puede provocar que más de uno salga corriendo y decida que la programación no es algo para él o peor aún, que decida que la detesta (como le pasa a muchos de esos estudiantes universitarios que te comentaba antes).  
Creo que hay lenguajes más adecuados para una primera toma de contacto con el mundo de la programación: como [Python](http://escueladeinformaticos.com/7-motivos-para-aprender-a-programar-con-python/), [JavaScript](http://escueladeinformaticos.com/si-quieres-crear-webs-interactivas-tienes-que-aprender-javascript/) (lenguajes que además, son de los más populares utilizados y demandados hoy en día).   EL lenguaje 'C++' es un lenguaje muy flexible y poderoso. Puede resultar complicado, pero se obtienen resultados buenos con su uso. Sim embargo esta quedando obsoleto, puesto a que nadie lo toma como un lenguaje bueno para poder empezar a programar. Java y demás lenguajes se están apoderando de todo, puesto a que los desarrolladores de esos lenguajes los hicieron más prácticos, eficaces y funcionales. Para el desarrollo de software, la fluidez del compilador, la optimización de la memoria virtual y el poder evitar la recolección de basura nativa es vital cuando se trabaja en algún proyecto, esto alivia la carga de trabajo humano y en muchas ocasiones evita errores, aunque también supone mayor carga de trabajo para el procesador.