



APRENDERAPROGRAMAR.COM

HISTORIA (RESEÑA) DE LA
PROGRAMACIÓN Y DE C.
LENGUAJES Y PARADIGMAS
(ESTRUCTURADA,
ORIENTADA A OBJETOS)
(CU00504F)

Sección: Cursos

Categoría: Curso básico de programación en lenguaje C desde cero

Fecha revisión: 2031

Resumen: Entrega nº4 del curso básico "Programación C desde cero".

Autor: Mario Rodríguez Rancel

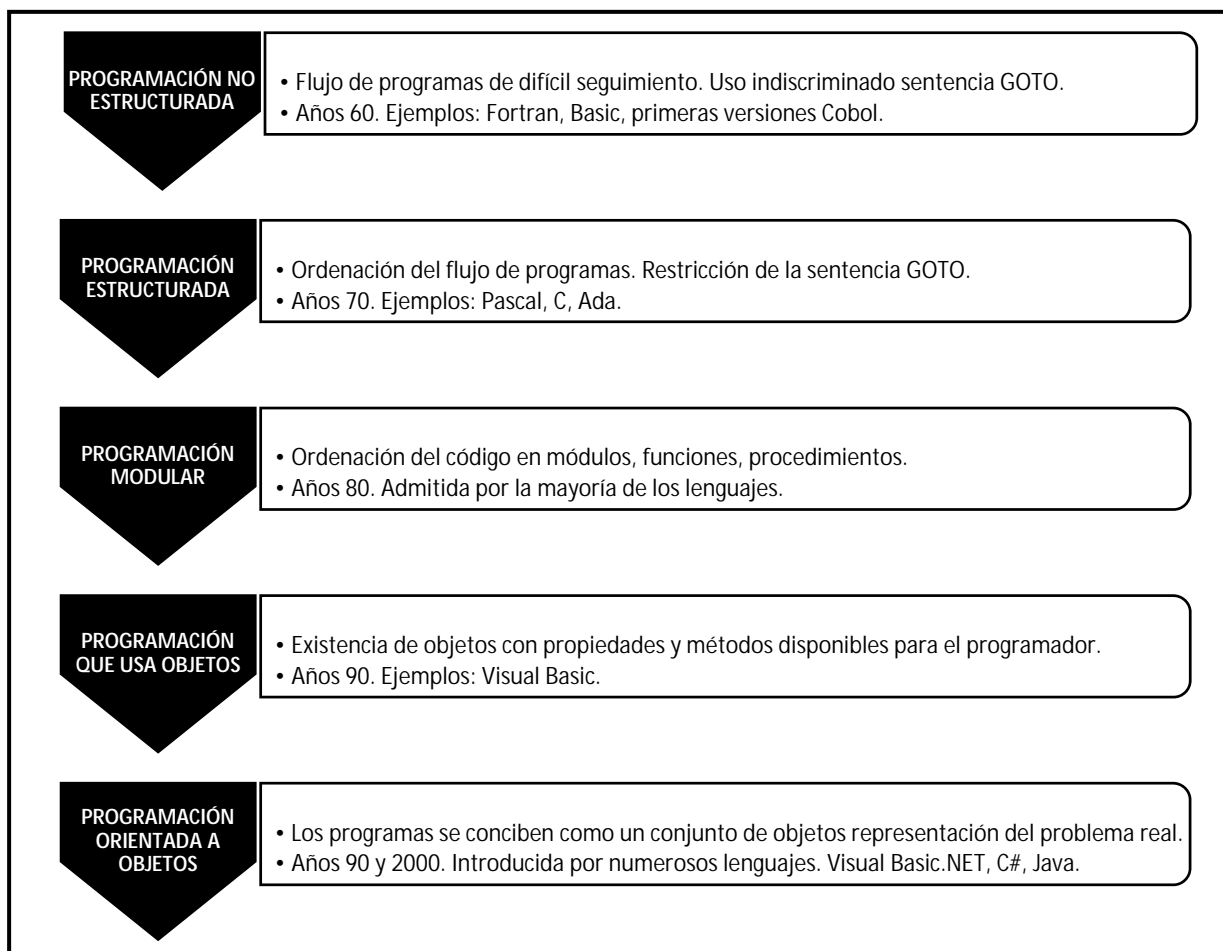
RESEÑA: HISTORIA DE LOS LENGUAJES DE PROGRAMACIÓN Y DE C

Probablemente habremos escuchado hablar de distintos lenguajes de programación: Pascal, Basic, Fortran, Cobol, C, C++, Visual Basic, Java, Perl, Python, etc. Podríamos hacer una lista de hasta 20 ó 30 lenguajes que son de amplio uso hoy en día (aunque algunos destacan por número de usuarios sobre los demás).



Los primeros lenguajes de programación fueron creados al mismo tiempo que aparecían los primeros computadores, hacia 1940. Se trataba de lenguajes denominados “de bajo nivel” cuya escritura era muy compleja. Posteriormente surgieron lenguajes que eran más fáciles de escribir y de interpretar. Se considera como primer lenguaje de alto nivel a Fortran, un lenguaje que surgió hacia finales de los años 50, diseñado para programación científica y cálculo numérico. Al contrario que otros lenguajes que cayeron en desuso, Fortran ha evolucionado y continuó siendo usado (incluso lo es hoy en día, aunque no sea tan popular como lo fue en el pasado).

De modo orientativo, este podría ser un esquema sobre la evolución de los lenguajes.



Este esquema es meramente orientativo: es imposible reflejar la diversidad de lenguajes y su evolución en un gráfico tan simplificado. También los datos relativos a fechas son orientativos.

Fortran puede decirse que introdujo los fundamentos de la programación moderna, al introducir muchos de los conceptos que hoy nos parecen básicos y que usan prácticamente todos los lenguajes de programación.

Tras Fortran surgieron otros lenguajes (COBOL, Algol60 y muchos otros).

No hay que suponer que lo único válido sea la programación orientada a objetos por ser lo más moderno. Al contrario, muchísima programación de la que se hace hoy en día se basa en lenguajes o código no orientado a objetos. Además, la misma programación orientada a objetos se basa en conceptos muy antiguos de programación (programación estructurada, modular, etc.).

Tener en cuenta que algunos lenguajes que nacieron en los años 60 han perdido vigencia y ya no se usan, mientras que otros se han ido modernizando y continúan usándose más o menos ampliamente, como es el caso de Cobol.

C es un lenguaje de gran importancia histórica, antecedente de algunos de los lenguajes más utilizados hoy en día (C++, Java), y que aún hoy sigue utilizándose y teniendo demanda laboral.

El lenguaje C fue desarrollado en la década de los 70 y constituye un lenguaje muy robusto usado como núcleo del sistema operativo Unix. C no admite la "orientación a objetos" y muchas veces se alude a él como un lenguaje poco amigable desde el punto de vista de que algunos aspectos de su sintaxis, elementos de programación que emplea (manejo directo de memoria) y otras cuestiones hacen que sea un lenguaje difícil de aprender en su totalidad. C++ fue una evolución de C desarrollada en los años 80. Introdujo el diseño orientado a objetos, pero manteniendo la compatibilidad con C. Esto suponía que C++ permitiera tanto el uso de la programación estructurada "tradicional" como la programación orientada a objetos. Además C++ mantuvo ciertas características de C que hicieron que la evolución fuera "parcial".

Como paso final en esta evolución estaría Java, un lenguaje que evoluciona a partir de C y C++, pero que elimina diversos aspectos de estos lenguajes y se constituye en un lenguaje definitivamente orientado a objetos, un lenguaje completamente nuevo. Java es un lenguaje popular y de amplio uso.

No te preocupes si no entiendes algunos de los conceptos que estamos nombrando como "orientación a objetos", "programación estructurada", etc. A medida que progreseemos con el curso, iremos introduciendo los conceptos que debemos estudiar así como ejemplos y ejercicios a realizar explicados paso a paso. Por ahora es suficiente saber que existen numerosos lenguajes de programación que han ido surgiendo a lo largo del tiempo.

CONSIDERACIONES PREVIAS SOBRE ESTE CURSO DE C

Este curso está pensado para aplicar los conceptos clave de la programación a un lenguaje concreto. Estudiar los fundamentos de programación sin realizar la aplicación en ordenador supone que muchos alumnos se sientan desencantados y realicen comentarios del tipo: "Está bien, hemos aprendido fundamentos, pero queremos ponernos delante de un ordenador y aplicarlos". Y precisamente esa es la filosofía de este curso. El objetivo es aplicar los conocimientos de programación que se suponen adquiridos previamente estudiando pseudocódigo, lógica y diagramas de flujo, a un lenguaje. Nótese que hay una diferencia entre esta propuesta y lo que sería un estudio en profundidad de una

metodología de programación o de un lenguaje concreto. Lo cual más que un curso requeriría un master, ya que los lenguajes y las posibilidades que ofrecen son muy extensas (es frecuente que los manuales exhaustivos de lenguajes tengan cientos o miles de páginas).

Puedes utilizar este curso para aplicar conocimientos previos y conocer aspectos básicos de programación estructurada y en concreto del lenguaje C, pero no recomendamos seguirlo si buscas un desarrollo en profundidad y metódico sobre estas materias. No vamos a entrar en definiciones ni en consideraciones abstractas. Tampoco vamos a seguir lo que sería un guión completo de estudio de todos los aspectos del lenguaje C: nos vamos a limitar a seguir el mismo guión que se usa en el curso de fundamentos de programación de aprenderaprogramar.com, pero en este caso "sobre un lenguaje". Es recomendable, aunque no imprescindible, haber seguido el curso de fundamentos (bases) de la programación de aprenderaprogramar.com antes de hacer este curso.

Dejaremos de lado los aspectos estéticos: el mundo de las formas de presentación, los colores, imágenes, efectos visuales, etc. es atractivo y parte más o menos importante de los programas, pero no nuestro objetivo. Nos vamos a centrar en los "esqueletos" de los programas: el código que hace que funcionen, no el que hace que sean bonitos.

Por último cabe hablar del lenguaje a utilizar. El conjunto de lenguajes y sus versiones constituyen una torre de Babel, un conglomerado de difícil seguimiento e interpretación. Este curso está orientado a aprender a programar. Es difícil elegir un lenguaje para aprender a programar, porque no existe acuerdo respecto a cuál es el ideal (quizás no existe un lenguaje ideal). Pero incluso después de elegir un lenguaje, que en nuestro caso va a ser C pero podría haber sido Visual Basic ó C++ u otro, hay que enfrentarse a la evolución de estos lenguajes y a las variantes dentro de una versión. Por ejemplo, si usáramos *Visual Basic 2012* nos encontraríamos con tres versiones: la estándar, la profesional y la empresarial. Ahí no acaba todo, al poco de estar usando esta versión comprobaríamos que ha salido *Visual Basic 2018*, que tendería a dejar obsoleta a la anterior, y al poco comprobaríamos que ha salido *Visual Basic 2025* y al poco... En definitiva, todo es un poco caótico visto desde fuera. Nuestra opinión consiste en dejar esa carrera loca, quizás necesaria, para los programadores profesionales y empresas, mientras que los usuarios podemos contentarnos con programas que funcionen, aunque sean un poco "antiguos". C es un lenguaje quizás más estable que otros, aunque también disponemos de muchas opciones diferentes respecto a posibles compiladores o entornos de programación a usar. Nosotros vamos a proponer el uso de un entorno de desarrollo concreto denominado Code::Blocks del cual existen versiones para Windows, Linux y Macintosh. No obstante, si lo deseas puedes usar otro entorno para programación C realizando las pequeñas adaptaciones que sean necesarias. En resumen, hay distintos entornos de desarrollo y versiones de lenguajes, que están en evolución constante, pero detrás de los aparentes cambios casi siempre se encuentra el mismo "fondo". A no ser que tengas motivos especiales, te recomendamos que uses Code::Blocks, un entorno que cuenta con miles de usuarios y que explicaremos a lo largo del curso.

Próxima entrega: CU00505F

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente:
http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=82&Itemid=210