



INSTITUTO TECNOLÓGICO DE MÉXICO CAMPUS PACHUCA

LENGUAJES Y AUTOMATAS

JORGE RAFAEL GARCIA SANDOVAL



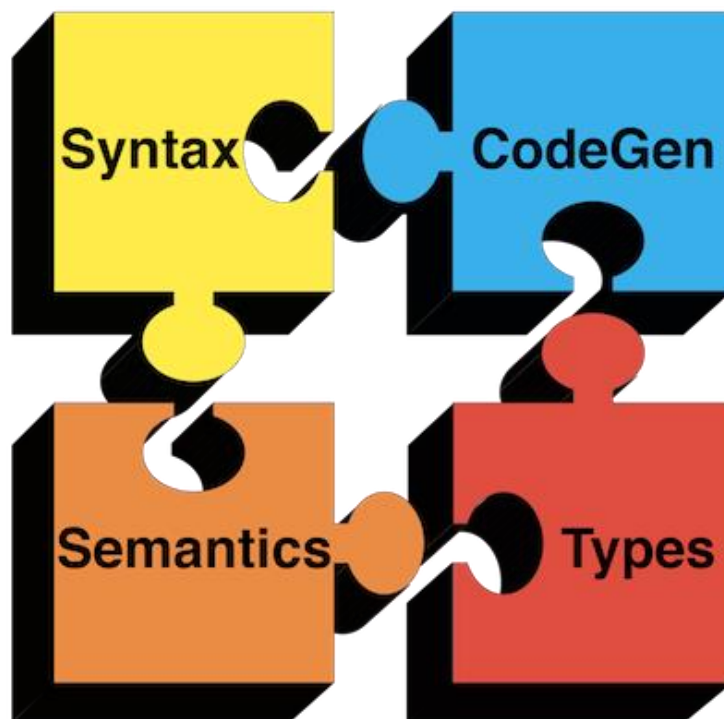
GRUPO: B

TAREA 1.3. HISTORIA COMPILADORES

INTRODUCCIÓN

Los compiladores, esos ingeniosos traductores de lenguajes de alto nivel a código de máquina, han sido una parte fundamental del desarrollo de la informática. A lo largo de las décadas, han evolucionado desde sus humildes comienzos hasta convertirse en herramientas sofisticadas que optimizan y transforman nuestro código. En esta línea del tiempo, trataremos de explorar los momentos clave que han dado forma a la historia de los compiladores.

Antes de los compiladores, los programadores escribían sus programas directamente en código de máquina o lenguaje ensamblador. Estos lenguajes eran difíciles de leer, propensos a errores y no portables entre diferentes arquitecturas de computadoras. La necesidad de un lenguaje más expresivo y reutilizable llevó al desarrollo de los compiladores.



HISTORIA COMPILADORES

CORRADO BÖHM Y SU TESIS DOCTORAL

LOGRO: BÖHM DESARROLLA UN LENGUAJE, UNA MÁQUINA Y UN MÉTODO DE TRADUCCIÓN PARA COMPILAR ESE LENGUAJE.

IMPACTO: DEFINE POR PRIMERA VEZ UN COMPILADOR EN SU PROPIO LENGUAJE, SENTANDO LAS BASES PARA LA TEORÍA DE COMPILADORES.



1951

1952



GRACE HOPPER ESCRIBE EL PRIMER COMPILADOR PARA EL LENGUAJE SISTEMA A-0

LOGRO: HOPPER CREÓ UN PROGRAMA QUE TRADUCÍA INSTRUCCIONES ESCRITAS EN LENGUAJE SISTEMA A-0 A CÓDIGO DE MÁQUINA.

IMPACTO: ESTE FUE EL PRIMER PASO HACIA LA CREACIÓN DE LENGUAJES DE ALTO NIVEL Y LA AUTOMATIZACIÓN DE LA TRADUCCIÓN DE CÓDIGO.

FORTRAN Y EL PRIMER COMPILADOR COMPLETO

LOGRO: EL EQUIPO DIRIGIDO POR JOHN W. BACKUS EN IBM PRESENTA EL PRIMER COMPILADOR COMPLETO PARA FORTRAN.

IMPACTO: FORTRAN SE CONVIRTIÓ EN EL PRIMER LENGUAJE DE ALTO NIVEL AMPLIAMENTE UTILIZADO, PERMITIENDO A LOS PROGRAMADORES ESCRIBIR PROGRAMAS MÁS LEGIBLES Y PORTABLES.



1957

1960



CONSOLIDACIÓN DE LOS COMPILADORES

LOGRO: LOS COMPILADORES SE ESTABLECEN GRADUALMENTE COMO HERRAMIENTAS ESENCIALES PARA LOS PROGRAMADORES.

IMPACTO: LOS LENGUAJES DE ALTO NIVEL COMO ALGOL, COBOL Y FORTRAN SE VUELVEN MÁS ACCESIBLES Y REUTILIZABLES.

GRAMÁTICAS Y ANALIZADORES SIN CONTEXTO

LOGRO: SE DESARROLLAN TÉCNICAS FORMALES PARA DESCRIBIR LA ESTRUCTURA SINTÁCTICA DE LOS LENGUAJES.

IMPACTO: ESTAS TÉCNICAS PERMITEN LA CONSTRUCCIÓN DE ANALIZADORES SINTÁCTICOS MÁS EFICIENTES Y PRECISOS.



1970

1980



GENERADORES DE ANALIZADORES Y METACOMPILADORES

LOGRO: HERRAMIENTAS COMO YACC Y LEX SIMPLIFICAN LA CREACIÓN DE COMPILADORES.

IMPACTO: LOS PROGRAMADORES PUEDEN GENERAR ANALIZADORES LÉXICOS Y SINTÁCTICOS AUTOMÁTICAMENTE A PARTIR DE ESPECIFICACIONES.

COMPILACIÓN CRUZADA Y OPTIMIZACIÓN DE COMPILADORES

LOGRO: LA COMPILACIÓN CRUZADA PERMITE COMPILAR PROGRAMAS PARA UNA ARQUITECTURA DIFERENTE DE LA DEL HOST.

IMPACTO: LA OPTIMIZACIÓN DE COMPILADORES MEJORA EL RENDIMIENTO DE LOS PROGRAMAS GENERADOS.



1990

2010



COMPILADORES ESPECÍFICOS PARA LENGUAJES MODERNOS

LOGRO: SE DESARROLLAN COMPILADORES ESPECÍFICOS PARA LENGUAJES COMO JAVA, PYTHON Y C#.

IMPACTO: ESTOS COMPILADORES PERMITEN LA EJECUCIÓN EFICIENTE DE PROGRAMAS ESCRITOS EN LENGUAJES POPULARES.

INVESTIGACIONES EN COMPILADORES CUÁNTICOS Y SISTEMAS EMBEBIDOS

LOGRO: LA COMUNIDAD CIENTÍFICA SIGUE EXPLORANDO LA CONSTRUCCIÓN DE COMPILADORES PARA COMPUTADORAS CUÁNTICAS Y SISTEMAS EMBEBIDOS.

IMPACTO: ESTOS AVANCES PODRÍAN REVOLUCIONAR LA INFORMÁTICA AL PERMITIR CÁLCULOS MÁS RÁPIDOS Y EFICIENTES.



2020

ACTUALIDAD

CONCLUSIÓN

La historia de los compiladores es una narrativa de innovación, desafíos técnicos y avances que han dado forma a la informática moderna. Desde los primeros pasos de estos, hasta las investigaciones actuales en compiladores cuánticos, estos ingeniosos programas siguen siendo fundamentales para los programadores en la actualidad. Ahora sé que durante la próxima vez que yo escriba un código en Java, Python o JavaScript, recordare que detrás de él hay un compilador que ha recorrido un largo camino desde los días del código de máquina hasta la actualidad y el futuro cercano.

BIBLIOGRAFÍA

Hopper, G. M. (1952). A Manual for the Automatic Sequence Controlled Calculator (ASCC). Harvard University, Cambridge, MA.

Backus, J. W., & Harlan, R. E. (1957). The FORTRAN Automatic Coding System. Proceedings of the Western Joint Computer Conference.

Sammet, J. E. (1969). Programming Languages: History and Fundamentals. Prentice-Hall.

Böhm, C. (1951). Grammatical Inference and Transformation of Languages. North-Holland Publishing Company.

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (1986). Compilers: Principles, Techniques, and Tools. Addison-Wesley.

Muchnick, S. S. (1997). Advanced Compiler Design and Implementation. Morgan Kaufmann.

Cooper, K. D., & Torczon, L. (2011). Engineering a Compiler. Elsevier.

Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). Compilers: Principles, Techniques, and Tools (2nd ed.). Addison-Wesley.