



UNIVERSIDAD AMERIKE

ESTRUCTURAS DE COMPUTADORES Y COMPILADORES

BRYAN SEBASTIÁN RUÍZ

SEBASTIAN MEJIA

INGENIERÍA EN CIBERSEGURIDAD

**COMPARATIVA DE LENGUAJES DE PROGRAMACIÓN, COMPILADOS E
INTERPRETADOS**

INTRODUCCIÓN

Los lenguajes de programación son herramientas fundamentales en el campo de la informática, diseñadas para comunicar instrucciones a una computadora y en términos generales, los lenguajes de programación se pueden clasificar en dos categorías principales: lenguajes compilados e interpretados.

DESARROLLO

COMPILADOS

Lenguaje C

El lenguaje C fue desarrollado por Dennis Ritchie de Bell Labs entre 1972 y 1973 para construir utilidades que se ejecutaban en el sistema operativo UNIX. Las primeras versiones de UNIX se escribieron en el lenguaje ensamblador o assembler language. Ese lenguaje de programación de bajo nivel se usaba cuando aún no había lenguajes de alto nivel y los recursos eran limitados. Con el lenguaje C el código del núcleo de UNIX se movió del ensamblador al lenguaje de nivel superior reduciendo líneas de código.

En 1985, apareció el sistema operativo Windows. Tanto Windows como Mac OS fueron escritos en C.

En 1991, el sistema operativo Linux se creó con el lenguaje C. Alrededor del 97% de las 500 supercomputadoras más potentes del mundo usan Linux.

Los sistemas operativos de dispositivos móviles iOS, Android y Windows Phone se basan en los núcleos de los sistemas macOS, Linux y Windows, y así funcionan a través del lenguaje C. Así, el lenguaje C constituye la base del trabajo de todos los dispositivos electrónicos.

Lenguaje C++

C++ es un lenguaje compilado que fue desarrollado en 1980 por Bjarne Stroustrup en los laboratorios At&T como una extensión orientada a objetos del lenguaje C.

La intención de su creación fue el extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

El impacto de C++ en la industria del software ha sido significativo, siendo utilizado en proyectos críticos donde se requiere un alto nivel de control sobre el hardware y el rendimiento. Su sintaxis clara y su capacidad para trabajar a bajo nivel lo hacen ideal para aplicaciones que necesitan un control preciso de los recursos del sistema.

Lenguaje Erlang

Se construyó a fines de los 80 en Ericsson para manejar conmutadores telefónicos. En ese momento, los sistemas de conmutación telefónica eran uno de los más complicados, como Internet actualmente. Por esta razón, el lenguaje utilizado para programarlos necesitaba admitir una alta concurrencia y cero tiempos de inactividad.

Después de revisar varias opciones de idioma existentes, tres personas de la empresa (Joe Armstrong, Robert Virding y Mike Williams) decidieron crear una propia.

En 1987, iniciaron los primeros experimentos con Erlang. Aquí suceden dos fases, la primera tenemos la construcción de prototipos de la funcionalidad PABX por parte de usuarios externos; segunda fase, dos años después reconstrucción de 1/10 del sistema MD-110 completo. Resultados: ¡10 veces mayores ganancias en eficiencia en la construcción en comparación con la construcción en PLEX!

Mike Williams escribió la primera máquina virtual basada en C y, un año después, se lanzó el primer proyecto comercial con un pequeño equipo de desarrolladores. El proyecto era un servidor de movilidad, que permitía a los usuarios de teléfonos inalámbricos DECT desplazarse por las redes de oficinas privadas.

Lenguaje Rust

Rust comenzó en 2006 como un proyecto paralelo de Graydon Hoare, un empleado de Mozilla. En poco tiempo, Mozilla vio el potencial del nuevo lenguaje y comenzó a patrocinarlo en 2009, antes de revelarlo al mundo en 2010.

La primera versión estable de Rust fue lanzada el 15 de mayo de 2015. En agosto de 2020, Mozilla despidió 250 empleados para reestructurar internamente la compañía a causa de la pandemia del Covid-19, entre ellos, el equipo Servo (un motor de renderizado web escrito en Rust). Esto creó preocupación sobre el futuro de Rust, pues varios de estos empleados contribuían activamente al lenguaje.

La siguiente semana, el equipo desarrollador central de Rust confirmó el impacto que habían causado los despidos al lenguaje, tomando la decisión de entregar los dominios y la propiedad completa del mismo a la fundación Rust y el 8 de Febrero de 2021 se anunció la creación formal de la fundación Rust conformada por AWS, Huawei, Google, Microsoft y Mozilla

INTERPRETADOS

Lenguaje PHP

Creado en 1994 por Rasmus Lerdorf, la primera encarnación de PHP era un conjunto simple de ficheros binarios Common Gateway Interface (CGI) escritos en el lenguaje de programación C. Originalmente utilizado para rastrear visitas de su currículum online, llamó al conjunto de scripts "Personal Home Page Tools", más frecuentemente referenciado como "PHP Tools".

Con el paso del tiempo se quiso más funcionalidad, y Rasmus reescribió PHP Tools, produciendo una implementación más grande y rica.

En septiembre de ese año, Rasmus amplió PHP y por poco tiempo abandonó el nombre de PHP. Ahora, refiriéndose a las herramientas como FI (abreviatura de "Forms Interpreter"), la nueva implementación incluía algunas de las funciones básicas de PHP tal y como la conocemos hoy. Tenía variables como las de Perl, interpretación automática de variables de formulario y sintaxis incrustada HTML. La sintaxis por sí misma era similar a la de Perl, aunque mucho más limitada, simple y algo inconsistente. De hecho, para embeber el código en un fichero HTML, los desarrolladores tenían que usar comentarios de HTML.

En 1997 y 1998, PHP/FI tenía un culto de varios miles de usuarios en todo el mundo. Una encuesta de Netcraft en mayo de 1998 indicó que cerca de 60,000 dominios reportaron que tenían cabeceras que contenían "PHP", indicando en efecto que el servidor host lo tenía instalado. Este número se correspondía con alrededor del 1% de los dominios de Internet del momento.

Lenguaje Ruby

Desde su liberación pública en 1995, Ruby ha atraído devotos desarrolladores de todo el mundo. En el 2006, Ruby alcanzó reconocimiento masivo, formándose grupos de usuarios activos en las ciudades más importantes del mundo y llenando las capacidades de las conferencias relacionadas a Ruby.

Ruby-Talk, la lista de correo más importante sobre el lenguaje Ruby creció hasta lograr un promedio de 200 mensajes por día en el 2006. Se ha reducido en los últimos años a medida que el tamaño de la comunidad movió la discusión de una lista central a muchos grupos más pequeños.

Inicialmente, Matz buscó en otros lenguajes para encontrar la sintaxis ideal. Recordando su búsqueda, dijo, “quería un lenguaje que fuera más poderoso que Perl, y más orientado a objetos que Python2”.

En Ruby, todo es un objeto, se le puede asignar propiedades y acciones a toda información y código. La programación orientada a objetos llama a las propiedades variables de instancia y las acciones son conocidas como métodos, la orientación a objetos pura de Ruby se suele demostrar con un simple código que aplica una acción a un número.

Lenguaje Python

La historia de Python empieza con Guido Van Rossum empezando su desarrollo en 1989 y empezando a implementarlo en febrero de 1991, momento en el que se publicó la primera versión pública: la 0.9.0.

Esta primera versión ya incluía clases con herencias, manejo de excepciones, funciones y una de las características principales de Python: funcionamiento modular. Esto permitía que fuese un lenguaje mucho más limpio y accesible para la gente con pocos conocimientos de programación. Una característica que se mantiene hasta el día de hoy.

La última gran actualización de la historia de Python se produjo en el año 2008 con el lanzamiento de la versión 3.0, que venía a solucionar los principales fallos en el diseño de este lenguaje de programación.

Aunque Python mantiene su filosofía en esta última versión, como lenguaje de programación ha ido acumulando formas nuevas y redundantes de programar un mismo elemento. De ahí la necesidad de nuevas versiones que eliminen estos constructores duplicados.

Hasta el año 2018, el desarrollo de este popular lenguaje de programación estaba dirigido personalmente por Van Rossum, pero decidió apartarse y, desde 2019, son cinco las personas que deciden cómo evoluciona y se desarrolla Python. Un consejo que se renueva de forma anual.

Lenguaje JavaScript

El origen javascript surge a partir de un navegador que actualmente no existe: NetScape. En 1995, Netscape se dio cuenta de que, cuando tenía algún documento HTML, no podía interactuar con lo que hacía el usuario. Lo único que podía hacer el navegador era esperar a recibir datos y enviarlos, a modo de formulario. Además, podía darle ciertos aspectos de diseño, como el alto y ancho, pero con muy poca versatilidad.

El siguiente momento importante en la historia de JavaScript es la creación del elemento XMLHttpRequest por parte de Microsoft en 1998. Este elemento viene a ser el padre de AJAX y nos permite cargar contenido o interactuar con el backend sin tener que volver a cargar la totalidad de la página web.

A lo largo de los años, se continuó hablando de JavaScript y, en el 2000, Douglas Crockford inventó el documento JSON, una forma de estandarizar objetos en JavaScript. En 2004, Gmail empezó a utilizar JavaScript e incorporó AJAX masivamente para hacer sus procesos más eficaces.

Llegados a este punto, JavaScript era tan útil que, en 2006, John Resig creó la librería JQuery, que nos permite reducir la complejidad de la sintaxis de JavaScript.

En 2009, no solo nace Google Chrome, sino que también nace el motor de JavaScript V8, que actúa como su intérprete. En el mismo año, Ryan Dahl se dio cuenta de que podía aislar el motor V8 de Chrome y lo usó para crear NodeJS, haciendo que la tecnología se separase del navegador.

Compilados

- **Compilación a Código Máquina:** El código fuente se traduce a código máquina mediante un compilador antes de su ejecución.
- **Eficiencia:** Los programas compilados suelen ser muy eficientes en términos de velocidad de ejecución y uso de recursos.
- **Errores en Tiempo de Compilación:** La mayoría de los errores son detectados durante el proceso de compilación.
- **Distribución:** El software se distribuye como binarios ejecutables específicos de la plataforma.

Interpretados

- **Ejecución en Tiempo de Ejecución:** El código fuente se ejecuta línea por línea mediante un intérprete.
- **Flexibilidad:** Permite una rápida iteración y desarrollo ágil.
- **Portabilidad:** El mismo código fuente puede ejecutarse en múltiples plataformas sin necesidad de recompilación.

CONCLUSIÓN

La elección entre lenguajes de programación compilados e interpretados depende en gran medida de las necesidades específicas del proyecto y del entorno de desarrollo. Para aplicaciones donde el rendimiento y el control sobre el hardware son cruciales, los lenguajes compilados son la opción preferida y con el avance continuo de la tecnología, es probable que sigamos viendo innovaciones que entre lenguajes compilados e interpretados ofrecen nuevas posibilidades y desafíos para los desarrolladores de software.

REFERENCIAS BIBLIOGRAFICAS

- <https://recluit.com/puntos-para-comenzar-a-utilizar-erlang/#:~:text=Erlang%20es%20un%20lenguaje%20funcional,Ericsson%20para%20manejar%20conmutadores%20telef%C3%B3nicos>.
- <https://recluit.com/descubre-la-historia-de-erlang/>
- <https://www.freecodecamp.org/espanol/news/lenguajes-compilados-vs-interpretados/>
- https://ferestrepoca.github.io/paradigmas-de-programacion/progconcurrente/tutoriales/Libro_Rust_2022_1/conociendo/rust/historia.html
- https://programacionpro.com/lenguaje-de-programacion-c-historia-y-evolucion/#google_vignette
- <https://www.sutori.com/en/story/historia-de-c--oSC1nm4mcz588PjEAMjjXZyo>
- <https://www.dongee.com/tutoriales/que-es-c-y-su-historia-en-la-programacion/>

- <https://ebac.mx/blog/que-es-lenguaje-c#:~:text=El%20lenguaje%20C%20fue%20desarrollado,lenguaje%20ensamblador%20o%20assembler%20language.>
- <https://www.php.net/manual/es/history.php.php>
- <https://www.ruby-lang.org/es/about/>
- <https://www.tokioschool.com/formaciones/cursos-programacion/python/historia/>
- <https://keepcoding.io/blog/la-historia-de-javascript/>
-