



Universidad Autónoma de Nuevo León
Facultad de Ingeniería Mecánica y Eléctrica

Lenguajes de Programación.

Actividad 7

Ing. José Antonio Juárez Covarrubias

Alumno: Jesús De Los Reyes Lárraga

Carrera: ITS

Grupo: Veranos

Hora: V4

Verano 2021

A lo largo de los años hemos visto como los lenguajes de programación han ido evolucionando de una manera vertiginosa. Si a inicios de la programación veíamos como las aplicaciones se desarrollaban con códigos binarios, hoy podemos abstraer la realidad de nuestro entorno con ayuda de herramientas muy completas como la programación orientada a objetos. En la actualidad existen diversas herramientas y metodologías de desarrollo de aplicaciones. Una de ellas, los lenguajes de scripting, ha ganado muchos adeptos alrededor del mundo entero. ¿Qué son los lenguajes de scripting? Los lenguajes de scripting son una subcategoría de los lenguajes de programación. Este tipo de lenguajes son interpretados, es decir, que para ejecutarlas instrucciones existe un programa o intérprete que se encarga de procesar una de las órdenes y producir los resultados deseados. Algunos ejemplos de este tipo de lenguajes son: JavaScript, Lisp, Perl, Python, Ruby, etc. En el otro caso, en los lenguajes que no son interpretados existe un compilador que toma esas instrucciones y genera un archivo ejecutable. Tenemos en esta categoría a C, C++, Java, Microsoft Visual Basic, etc. Recientemente hemos visto dos lenguajes de scripting orientados a objetos que han tenido una gran acogida por la comunidad de programadores: Ruby y Groovy.

Ruby

En 1993, un científico de la computación, el japonés Yukihiro Matsumoto, inicia el desarrollo de un novedoso lenguaje de scripting orientado a objetos, basado en Python. Este lenguaje, bautizado con el nombre de Ruby, en la actualidad está considerado dentro de los 10 mejores lenguajes de programación por TIOBE¹.

Existen herramientas que facilitan la tarea de escribir compiladores o intérpretes informáticos. Estas herramientas permiten generar el esqueleto del analizador sintáctico a partir de una definición formal del lenguaje de partida, especificada normalmente mediante una gramática formal y barata, dejando únicamente al programador del compilador la tarea de programar las acciones semánticas asociadas.

En ciencias de la computación, intérprete o interpretador es un programa informático capaz de analizar y ejecutar otros programas. Los intérpretes se diferencian de los compiladores o de los ensambladores en que mientras estos traducen un programa desde su descripción en un lenguaje de programación al código de máquina del sistema, los intérpretes solo realizan la traducción a medida que sea necesaria, típicamente, instrucción por instrucción, y normalmente no guardan el resultado de dicha traducción.}

En 1945, el matemático y químico John (Janos) von Neumann presenta los principios generales que debe seguir una máquina de propósito general.

El primer lenguaje en el que se programaron los ordenadores fue el propio del procesador, es decir, instrucciones análogas a las presentes en la máquina de Von

Neumann. Sin embargo era necesario tener en cuenta los detalles propios de la máquina para poder realizar cualquier cálculo y además era muy tedioso introducir el programa

en el ordenador.

Usando un intérprete, un solo archivo fuente puede producir resultados iguales incluso en sistemas sumamente diferentes (ejemplo. una PC y una PlayStation 4). Usando un compilador, un solo archivo fuente puede producir resultados iguales solo si es compilado a distintos ejecutables específicos a cada sistema.

Los programas interpretados suelen ser más lentos que los compilados debido a la necesidad de traducir el programa mientras se ejecuta, pero a cambio son más flexibles como entornos de programación y depuración (lo que se traduce, por ejemplo, en una mayor facilidad para reemplazar partes enteras del programa o añadir módulos completamente nuevos), y permiten ofrecer al programa interpretado un entorno no dependiente de la máquina donde se ejecuta el intérprete, sino del propio intérprete (lo que se conoce comúnmente como máquina virtual).

Para mejorar el desempeño, algunas implementaciones de algunos lenguajes de programación pueden interpretar o compilar el código fuente original en una forma intermedia más compacta, y después traducir eso al código de máquina (ej. Perl, Python, MATLAB, y Ruby). Algunos aceptan los archivos fuente guardados en esta representación intermedia (ej. Python, UCSD Pascal y Java).

En la actualidad, uno de los entornos más comunes de uso de los intérpretes es en los navegadores web, debido a la posibilidad que estos tienen de ejecutarse independientemente de la plataforma.

Groovy un lenguaje de scripting orientado a objetos y mantiene una sintaxis muy similar a Java. Recientemente se liberó la versión 1.0 de dicho lenguaje, bajo licencia GNU. Las aplicaciones desarrolladas en Groovy pueden ser ejecutadas de forma normal por cualquier máquina virtual de Java. Potencialidades de un lenguaje de scripting orientado a objetos Este tipo de lenguaje tienen grandes

ventajas frente a la forma de programar clásica. Algunas de las características más sobresalientes son:

- No se requiere ciclo de compilación. – En los lenguajes de scripting no es necesario compilar el código fuente, sí lo creamos el programamos lo ejecutamos. Programación más simple. – No se requieren declarar los tipos de las variables, ni usar paréntesis para llamar a los métodos. Los arreglos se declaran de forma directa (sin necesidad de instancias).
- Orientación a objetos. – Al igual que otros lenguajes, también se pueden crear módulos, clases e interfaces.
- Soporte para manejo de hilos. – Se pueden crear hilos multitarea al igual que en C++ o Java.
- Control de excepciones. – Se tiene un robusto sistema para control y recuperación de errores.

De esta manera, podemos ver que esta nueva generación de lenguajes busca que la programación sea más sencilla, que las instrucciones se acerquen cada vez más al lenguaje natural y que los tiempos de desarrollo de aplicaciones se reduzcan de forma drástica.