

Ruby: Lenguaje de Programación para Sistemas Distribuidos

Nota de Divulgación

Lic. Héctor de Jesús Carlos Pérez (1), C.L.I. José Alfredo Alonso Esquivel (2) (1) Miembro del Cuerpo Académico de Sistemas Distribuidos del Instituto Tecnológico de Aguascalientes

(2) Alumno de la Carrera de Licenciatura en Informática del Instituto Tecnológico de Aguascalientes

Departamento de Sistemas y Computación del Instituto Tecnológico de Aguascalientes, Av. A. Lopez Mateos 1801, Fracc. Bona Gens, Aguascalientes, Ags., C.P. 20256 Tel. (449) 9105002, Fax (449)9700423
hjcarlos@hotmail.com, chokmah19@hotmail.com

Resumen

El desarrollo de los lenguajes de programación se está orientando a la generación de código distribuido. Mediante este tipo de código se puede hacer uso de programas, recursos, objetos y sistemas que no se tienen en el equipo local, pero que para el usuario final pareciera como si realmente estuvieran en su equipo.

Gran parte de los sistemas distribuidos requieren que los usuarios adquieran los respectivos permisos y licencias.

En este trabajo se describen las características y ventajas de Ruby, un lenguaje que permite generar este tipo de sistemas distribuidos de manera relativamente sencilla, y además presenta la ventaja de ser un lenguaje de código abierto y libre de distribución y mejoras por parte del usuario-programador.

Palabras clave

Sistema distribuido, Ruby, Código abierto.

Abstract

The development of programming languages is addressing to the generation of distributed code. Through this type of code it is possible to use programs, resources, objects and systems that not exist in local equipment, but they seems as if really they were in the equipment of local station.

Most of distributed systems need users acquire the permissions and licenses.

This work describes characteristics and advantages of Ruby, which is a language that allows to generate this type of distributed systems in a relatively simple way. Additionally has the advantage of being a free language and open source of distribution and improvements for the programmer and users.

Key words

Distributed system, Ruby, Opened source.

Introducción

De acuerdo a Wikipedia Ruby *"es un lenguaje de programación reflexivo y orientado a objetos creado por el programador japonés Yukihiro "Matz" Matsumoto en 1993. Combina una sintaxis inspirada en Perl con características de programación orientada a objetos similares a Smalltalk. Comparte también funcionalidad con otros lenguajes de programación como Python, Lisp, Dylan y CLU. Ruby es un lenguaje de programación interpretado y su implementación oficial es distribuida bajo una licencia de software libre"*.

La popularidad de este lenguaje radica en que reúne las características de los mejores lenguajes (Java, C, Python, entre otros) dentro de un relativamente sencillo lenguaje de programación. Dichas características son:

- Es poderoso: combina el poderío de los lenguajes orientados a objetos con la conveniencia de los lenguajes basados en scripts, como Perl. Los programas en Ruby son compactos, pero aún así muy mantenibles.
- Simple: La sintaxis y la semántica son intuitivas y muy limpias. No hay ningún 'caso especial' por tomar en cuenta. Por ejemplo, variables como integer, string, inclusive NULL son todas objetos.
- Transparente: Ruby libera de las limitaciones de espacio. No requiere compilar cada código para cada mejora o agregado de un nuevo módulo.
- Posee alta disponibilidad: con Ruby se pueden generar aplicaciones y correrlas sin ningún problema bajo Unix, Linux, Windows o sistemas especializados como lo es el BeOS, entre otros.

Características

- Posibilidad de realizar directamente llamadas al sistema operativo.
- Potentes operaciones sobre cadenas de caracteres y expresiones regulares.

- Retroalimentación inmediata durante el proceso de desarrollo.
- Las variables son de tipo dinámico.
- La gestión de la memoria es automática.
- Modelo de procesamiento de excepciones.
- Carga dinámica.
- Hilos.
- Clases, herencia y métodos innatos.
- Iteradores y cierres.

Ruby distribuido

Se pueden crear objetos independientes para poder llamarse después, (por ejemplo .dll). Si a estas características le agregamos que se pueden transmitir estos objetos a diferentes procesos, todo esto en conjunto con las tecnologías de comunicaciones, permite obtener los sistemas distribuidos.

Utilizando las características de Distributed Ruby Library, este puede ser usado como servidor, como host o inclusive como ambos. Actuando como servidor es un origen para los objetos; actuando como cliente pareciera que los objetos se obtienen localmente, pero realmente el código se está ejecutando en alguna otra parte del sistema distribuido.

El servidor comienza un servicio o demonio asociando un objeto con un puerto determinado. Los hilos se crean dentro del servidor para manejar requisiciones que provengan de dicho puerto.

Ruby y la Web

Con este lenguaje se pueden crear gran cantidad de tecnologías. Algunas de ellas son:

- Servidor SMTP.
- Demonios FTP.
- Servidores Web.

Inclusive se puede usar para trabajos más específicos, como programar CGI's o como reemplazo para PHP.

Los CGI's son simplemente páginas Web, pero estas son resueltas por el servidor; el usuario final no se da cuenta de todos los procesos (accesos a bases de datos, manejo de formularios, etc.) que se generan en el servidor; el usuario final sólo recibe los resultados, en comparación con las páginas ASP, PHP en las cuales se puede tener acceso a ciertas partes del código que se utiliza para crear las páginas; un ejemplo de esto es simplemente dar clic derecho en la página y seleccionar 'ver código fuente'.

Como se menciono anteriormente, Ruby se puede usar para generar CGI's. Para escribir scripts en CGI

con Ruby, simplemente se necesita mandar un encabezado al inicio de la programación, como el siguiente:

```
#!/usr/bin/env ruby
Print "HTTP/1.0 200 OK\r\n";
Print "Content-type: text/html \r\n\r\n";
Print "<html><body>Hello
World!</body></html>";
```

Con esto basta para comenzar a generar CGI's. También de una manera muy sencilla e intuitiva, para aquellos familiarizados con los lenguajes mencionados al principio de este artículo, se pueden crear cualquier tipo de elementos Web (para el caso de los CGI's) como son:

- Formularios
- Cookies
- Sesiones

Con Ruby, se pueden observar todos los objetos vivos dentro del bloque de código que se tiene en cierto momento. Para ejemplificar esto, tomemos el siguiente segmento de código:

```
a = 102.7
b = 95.1
ObjectSpace.each_object(Numeric)
{|x| p x}
```

Nos muestra en pantalla:

```
95.1
102.7
2.718281828
3.141592654
```

Una nota curiosa acerca de este lenguaje es que originalmente era el nombre de un tipo de letra británico de 5,5 puntos usado para anotaciones en documentos impresos. En japonés, el significado de la palabra cambió para referirse a los caracteres *furigana*.

Conclusiones

El lenguaje Ruby viene a reemplazar muchos otros lenguajes, siendo éste capaz de cumplir con todas las características de estos, e inclusive llegando hasta donde los demás lenguajes no pueden.

En México no hay gran aceptación por los sistemas de código abierto, situación que debe de cambiar, ya que éste ofrece grandes ventajas por sobre los otros sistemas: amplia disponibilidad, bajo costo si no es que nulo, robustez, capacidad de crecimiento, soporte por bastantes grupos de programadores en todo el mundo, más seguridad que los demás sistemas operativos,

capacidades de sistemas operativos distribuidos, en fin, el uso de estas herramientas beneficia tanto a las empresas que deseen implementar tecnología de punta así como a programadores, usuarios finales y administradores. Ruby, aunque requiere cierto nivel de experiencia en programación, es una excelente herramienta para comenzar la migración de otros sistemas operativos hacia los de código abierto.

Referencias

- [1] <http://www.groovie.org/articles/2005/10/31/the-wacky-world-of-ruby>
- [2] <http://www.rubycentral.com/articles/index.htm>
- [3] http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Ruby

Agradecimientos

Queremos agradecer al Cuerpo Académico de Sistemas Distribuidos del Instituto Tecnológico de Aguascalientes, así como a todos los colaboradores que brindaron su apoyo, los servicios y recursos necesarios para llevar a cabo la publicación de este trabajo.

Artículo recibido: 5 de octubre de 2006

Aceptado para publicación: 18 abril de 2007