

Cobol vs Ruby

Yaneth Aquino,,Bianca Chura ,Adnner Esperilla y Johanna Torres

June 6, 2020

Abstract

La comparación de lenguajes de programación es un tema común de discusión entre los ingenieros de software. Pocos lenguajes son lo suficientemente populares como para que sean utilizados por más de unas pocas personas o que encuentren su nicho en la investigación o la educación; Pero los programadores profesionales pueden utilizar fácilmente docenas de idiomas diferentes durante su carrera. Se implementan cada año con el fin de mantenerse al día con la programación cambiante paradigmas, evolución del hardware, etc.

I. INTRODUCCION

Los primeros lenguajes de programación de alto nivel fueron diseñados durante la década de 1950. Desde entonces, la programación idiomas han sido un área de estudio fascinante y productiva . Miles de programación diferente se han creado lenguajes, principalmente en el campo de la informática, y cada año se han creado muchos más; son diseñados, especificados e implementados con el propósito de estar al día con la evolución de la programación paradigmas (por ejemplo, imperativos, orientados a objetos, orientados a aspectos, reflexivos y funcionales, por nombrar algunos). Mientras que algunos lenguajes de programación gozan de popularidad en todo el mundo y se utilizan comúnmente para desarrollar grandes, aplicaciones de nivel empresarial (por ejemplo, C, C++, C, Java, PHP o Perl); otros sólo son utilizados por un número menor de personas o están más orientados a la academia para la investigación y la educación (por ejemplo, OCaml, Haskell, Scheme, OZ, o Scala).

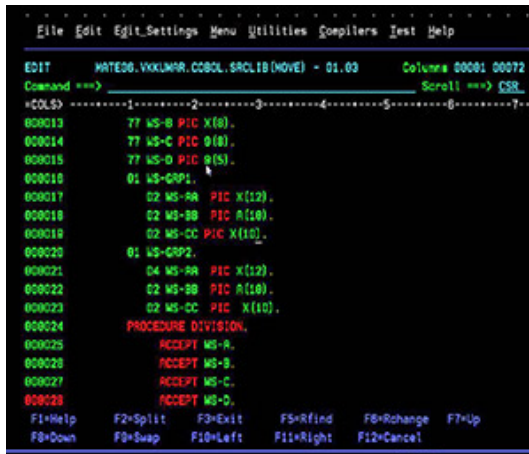
II. OBJETIVOS

- Establecer diferencias entre Cobol y ruby ,en las paradigmas de programacion.

III. DESARROLLO

i. COBOL

- Es uno de los lenguajes de programación más antiguos (creado en 1959). Ha dominado la aplicación de software desarrollo durante décadas y sigue siendo uno de los lenguajes de programación utilizados hoy en día en el software de negocios. COBOL presenta diferentes ventajas, como simplicidad, portabilidad y mantenimiento, también COBOL se considera muy poderoso en el procesamiento de datos masivos para las empresas Aplicaciones. COBOL ha sufrido varias mejoras (revisiones de 1968, 1974, 1985 y 2002) con el fin de ponerse al día con las nuevas tendencias de programación. La última encarnación de COBOL admite características orientadas a objetos (todavía no hay ningún compilador compatible disponible) y también mantiene la compatibilidad con versiones anteriores.



ii. Ruby

Ruby es un lenguaje de programación dinámico, reflexivo y orientado a objetos de propósito general que combina sintaxis inspirado por Perl con características similares a Smalltalk. Ruby se originó en Japón a mediados de la década de 1990 y fue desarrollado y diseñado por Yukihiro "Matz" Matsumoto. Ruby es un lenguaje de programación puro orientado a objetos con una sintaxis súper limpia que hace que la programación sea elegante y divertida. Ruby combina con éxito Smalltalk's elegancia conceptual, la facilidad de uso y aprendizaje de Python, y el pragmatismo de Perl. Ruby ha comenzado a convertirse en popular en todo el mundo en los últimos años como más libros en inglés y documentación se han convertido en Disponible. Ruby soporta múltiples paradigmas de programación, incluyendo funcionales, orientados a objetos, imperativos, y reflexivo. También cuenta con un sistema de tipo dinámico y gestión automática de la memoria



iii. Diferencia entre Cobol vs Rubi

• COBOL

- Es un lenguaje prototípico Aspect-COBOL para diseñar una extensión a COBOL para programación orientada a aspectos.
- COBOL no puede hacer programación funcional.
- COBOL es principalmente un lenguaje imperativo.
- Muy conocido en secuencias de comandos por lotes, COBOL puede automatizar las tareas llamando a programas externos como el controlador de la base de datos.

-Generalmente los programas COBOL se ejecutan a través de la línea de comandos, pero hay algunas herramientas disponibles para crear interfaces gráficas (que podría utilizarse para la creación de prototipos de interfaz de usuario).

• RUBY

- Ruby también cuenta con algunas extensiones: AspectR-Fork y Aquarium siendo los más popular y generalizada.
- Los bloques, procs y lambdas de Ruby prestan a sí mismos muy bien a una programación funcional Estilo. - Ruby carece de dos aspectos importantes para programación funcional (coincidencia de patrones y evaluación diferida), pero sus instalaciones, como bloques, lambdas, y el hecho de que todo se evalúa como una expresión que admite un estilo de programación funcional .
- Los bloques, procs y lambdas de Ruby prestan a sí mismos muy bien a una programación funcional Estilo. - Ruby carece de dos aspectos importantes para programación funcional (coincidencia de patrones y evaluación diferida), pero sus instalaciones, como bloques, lambdas, y el hecho de que todo se evalúa como una expresión que admite un estilo de programación funcional .
- Hay algunas opciones para ejecutar Batch puestos de trabajo en Ruby, y

también en los ferrocarriles Marco de referencia. El módulo `ActiveRecord::Batches::ClassMethods` permite procesamiento de un gran número de registros y deportes dos métodos principales para hacer scripting por lotes

- Existen varias bibliotecas para permitir que Ruby interacción gráfica del usuario (GUI).

IV. CONCLUSIONES

Es importante tomar decisiones tecnológicas en el momento adecuado y por las razones correctas. Buen negocio las decisiones proporcionan a las buenas personas las herramientas de apoyo adecuadas para que puedan producir buenos productos. Cuando se trata de desarrollo de software, lidiar con problemas de lenguaje difíciles de frente es una requisito para el gerente visionario de hoy. Cuando se combina con otra ingeniería de software consideraciones, una buena decisión de lenguaje puede apoyar el desarrollo de software rentable sistemas que, a su vez, proporcionan un soporte empresarial valioso y confiable.

REFERENCES

Al-Qahtani, S. S., Pietrzynski, P., Guzman, L. F., Arif, R., Tevoedjre, A. (2010). Comparing selected criteria of programming languages java, php, c++, perl, haskell, aspectj, ruby, cobol, bash scripts and scheme revision 1.0-a team cplgroup comp6411-s10 term report. arXiv preprint arXiv:1008.3434.