

Cobol vs Ruby

Yaneth Aquino, Bianca Chura, Adnner Esperilla y Johanna Torres

June 13, 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. se han creado lenguajes de programación, 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).

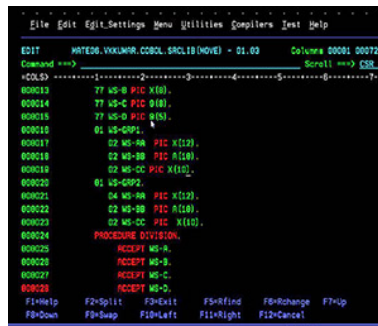
II. OBJETIVOS

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

III. DESARROLLO

i. COBOL

- Es uno de los lenguajes de programación más antiguos. 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 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. 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



```

Lenguaje Ruby
class Persona
  def initialize(nombre, fecha_nac) # Constructor
    @nombre, @fecha_nac = nombre, fecha_nac #Atributos privados
  end

  def printNombre() #Métodos de acceso
    return @nombre
  end

  def edad()
    edad = (DateTime.now() - @fecha_nac).to_i/364.5
    return edad
  end

  def to_s
    "Nombre:#{@nombre}, edad:#{@edad}" # Formato en String del objeto
  end
end

p = Persona.new("Cristian", Date.new(1992, 03, 17))
puts p.edad

```

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. (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.

• 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.
- 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).

Lenguaje	Uso Previsto	Imperativo	Orientado a objetos	Funcional	Procesal	Generico	Reflexivo	Evento conducido	Otro Paradigma	Estandares
Cobol	Aplicación, negocios	si	si	-	si	-	-	-	-	ANSI X3.23 1968, 1974, 1980, ISO/IEC 15926:1985, 2002, 2014
Ruby	Aplicación, scripting, web	si	si	si	-	-	si	-	orientado a aspectos	2013 (ISO X 3017), 2012 (ISO/IEC 30170)

iv. Funciones de métricas COBOL

- Operación repetible en un conjunto específico de archivos
- Métricas calculadas a nivel de párrafo, sección, división y archivo:

Líneas de código fuente no comentadas

Numero de metodos

Complejidad Ciclomática

Máxima anidación de bucle

Máxima anidación condicional

Medidas de Halstead (volumen, dificultad y esfuerzo) Índice de Mantenibilidad del Instituto de Ingeniería de Software (SEI)

- Las métricas de anidación en el peor de los casos señaladas a un archivo / línea específicos
- Resúmenes por sección / división / archivo
- Informe de texto
- Informe XML, reformateable en cualquier forma deseada

[illegible]

- rotación
- la complejidad costo
- la calificación .
- Ejemplo:para proporcionar un informe de calidad de su código Ruby

Churn : número de veces que se cambió un archivo

Complejidad : cantidad de dolor en su código

Calificación : la calificación asignada a un archivo

Code					
Rating ↓	Name	Churn	Complexity	Duplication	Smell
C	Schema	7	63	110	Schema
A	ApplicationController	4	2	0	Controller
A	CommentsController	7	33	0	Controller
A	OmniauthCallbacksController	1	8	0	Controller
A	PinController	10	44	0	Controller
A	RelationshipsController	2	13	0	Controller
A	UserController	11	66	0	Controller

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 un 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.

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.