



PARADIGMA REFLEXIVO

Integrantes:

Simon Correa Rios

Juan Felipe Zuleta Moreno

Alejandro Naranjo Cataño

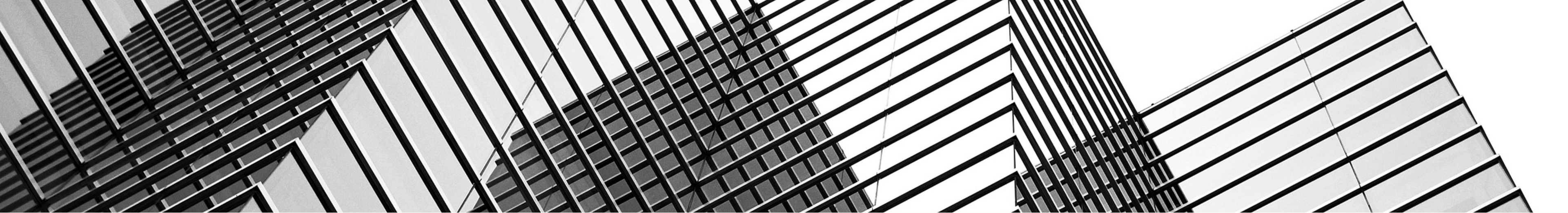
Sara Mesa Gómez

Jose Miguel Correa Sanchez

Juan Esteban Ballesteros Sánchez

Jhon Esteban Palacio Mazo





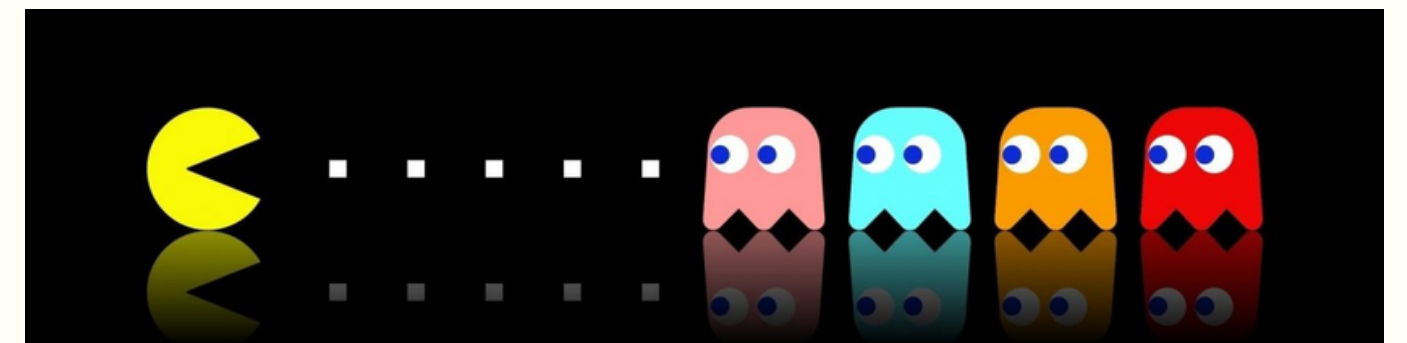
DESCRIPCIÓN

El paradigma reflexivo en programación se refiere a la capacidad que tiene un programa para observar y, opcionalmente, modificar su propia estructura de alto nivel. En otras palabras, es la habilidad de un programa para examinarse a sí mismo y adaptarse según las circunstancias.

Puntos clave

Observación dinámica

Modificación opcional



PROBLEMAS PRESENTES

- A pesar de que este paradigma va ligado con la creación de sistemas autónomos, hay que tener en cuenta que no siempre es el más eficiente.
- El razonamiento es una característica que resulta difícil de lograr artificialmente, en consecuencia el lograr que un sistema evalúe su entorno y tome decisiones a partir de este no es una tarea sencilla.
- Posee limitaciones con ciertos lenguajes de programación ejm (Visual Basic, C#)
- No siempre resulta ser el más efectivo
- (lo anterior lo podemos ver en la tabla a continuación)

Tabla 2. (a) Resultados para la Prueba 1. Instanciamiento de una clase.

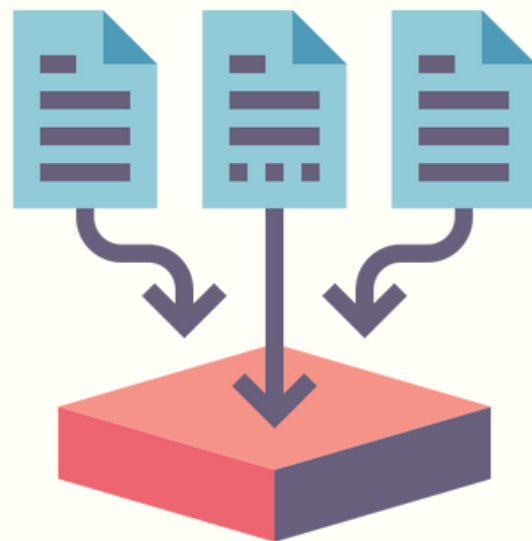
Número posible de clases a instanciar	1	20	40	60	80	100	Promedio
Tiempo de ejecución programa no reflexivo (ns)	875179	827368	849074	847928	854944	851472	850994
Tiempo de ejecución programa reflexivo (ns)	1357474	1341081	1415647	1497070	1466793	1403937	1413667



APLICACIONES

Ya sabemos que el paradigma reflexivo permite inspeccionar y modificar componentes del código al momento de ejecución, ahora ¿Para qué nos sirve esto?

- Es usado para crear librerías de software genéricas, para mostrar información.
- Procesar diferentes formatos de información
- La serialización y deserialización de datos.
- Permite la creación de programas autónomos.
- Es una estrategia clave para la metaprogramación.
- Agrupar y des agrupar datos.
- Descubrir y modificar construcciones de código fuente como objetos de categoría superior al ejecutar el programa.
- Crear objetos de prueba (mock objects) al momento de realizar pruebas de un programa.



BENEFICIOS

Flexibilidad

La reflexión permite a los desarrolladores crear aplicaciones más flexibles y adaptables. Puedes analizar y manipular clases, métodos y atributos en tiempo de ejecución, lo que facilita la creación de sistemas dinámicos



Autoinspección y depuración

El paradigma reflexivo permite a un programa observar y modificar su propia estructura en tiempo de ejecución.

Inspección de metadatos

Puedes obtener información sobre las clases, interfaces, métodos y propiedades de tu programa. Esto es útil para herramientas de desarrollo, frameworks y bibliotecas

Generación de código dinámico

Puedes crear y modificar clases y métodos en tiempo de ejecución.



LENGUAJES QUE LO SOPORTAN

Hay diferentes lenguajes que soportan la programación reflexiva, algunos mejor que otros, por lo que podemos considerarlos según el nivel de reflexión que soportan. Los resultados a continuación fueron obtenidos de un experimento realizado para comprobar la efectividad de la aplicación de este tipo de paradigma.

Muchos lenguajes de programación ofrecen soporte a funciones de reflexión básicas, pero si queremos enfocarnos en la implementación de este paradigma, los lenguajes C++, Java y SmallTalk, son los indicados, junto con PHP y PL/SQL que son segundos al nivel de reflexión que soportan los demás lenguajes.

Tabla 1. Lenguajes de programación y nivel de reflexión soportado.
Construcción propia

Lenguaje	Nivel de reflexión soportado	Observación
ActionScript 3	Primero	Flash ofrece el paquete flash.utils el cual provee funciones de reflexión esenciales. Este paquete devuelve un archivo XML con información del objeto (RIA Solutions, 2012; Adobe, 2012).
C#	Primero	C# tiene de forma nativa métodos como Assembly.GetExecutingAssembly() e Invoke(), entre otros; que permiten obtener información de los objetos instanciados (Microsoft-a, 2005).
C++	Tercero	En forma similar al API BCEL de Java, se puede incorporar código a los archivos generados luego de compilado el código por medio de clases especializadas desarrolladas por el programador. También está el API Reflection y librerías como LibReflection (Achilleas, 2004) y C++ reflection (Raiser, 2005).
Java	Tercero	Se puede lograr por medio de la API BCEL (BCEL, 2013; APACHE-a, 2013; APACHE-b, 2013) la cual permite incorporar código al BYTECODE.
Perl	Primero	Perl ofrece de forma nativa métodos de reflexión esenciales (Warden, 2012).
PHP	Segundo	PHP incluye una API para el manejo de reflexión. Se puede lograr que el usuario ingrese código en tiempo de ejecución (PHP, 2013).
PL/SQL	Segundo	PL/SQL de Oracle permite ejecutar código enviado en tiempo de ejecución mediante la sentencia EXECUTE IMMEDIATE (Oracle-a, 2013; Oracle-b).
Ruby	Primero	Ruby ofrece de forma nativa métodos de reflexión esenciales (Thomas, et al., 2004).
Smalltalk	Tercero	Smalltalk tiene meta-objetos, los cuales permiten que los objetos modifiquen su código (Black, et al., 2009).
Visual Basic	Primero	Visual Basic tiene de forma nativa métodos como GetType(), que permiten obtener el tipo de las variables declaradas (Microsoft-b, 2012; Patison, 2013).

COMUNIDADES Y FOROS WEB

GitHub

Los proyectos de código abierto alojados en GitHub a menudo utilizan la reflexión para realizar tareas como la generación dinámica de código, la inspección de clases y la manipulación de metadatos.



Python Subreddit (r/python):

Python es conocido por su soporte incorporado para la reflexión. En su comunidad de Reddit puedes encontrar una amplia variedad de discusiones sobre temas relacionados con Python y sus usuarios suelen debatir sobre el uso avanzado de este lenguaje, en estas la presencia de este paradigma resulta notoria.



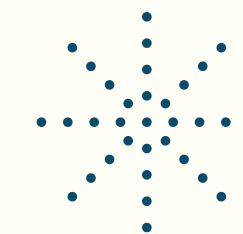
Ruby Subreddit (r/ruby)

Es una comunidad en Reddit dedicada a discutir sobre Ruby y Ruby on Rails. framework que utiliza la reflexión para mapear objetos de base de datos a clases de Ruby y viceversa, lo que facilita la creación rápida de aplicaciones web sin la necesidad de escribir código repetitivo.





Referencias



Moreno, F., Jiménez, J., & Castañeda, S. (2014). Una propuesta para la clasificación de la programación reflexiva orientada al desarrollo de sistemas autónomos. Ingeniería y Competitividad, 16(2), 91-104.

(Reflection (Computer Programming) - Ilmu Komputer | Wiki eduNitas.com, n.d.)

Wikipedia contributors. (2024, February 1). Reflective programming. Wikipedia.
https://en.wikipedia.org/wiki/Reflective_programming

Hunt, A., & Thomas, D. (2006). The Pragmatic Programmer's Guide. <http://phrogz.net/programmingruby/ospace.html>



