

Selección de Lenguajes Orientados a Objetos para un estudio comparativo y análisis de rendimiento

¹BENEDETTO Marcelo Gabriel, ¹CARABIO, Ana Lía Ramona, ¹ALVEZ Carlos Eduardo, ¹FERNÁNDEZ, Miguel Antonio, ¹ETCHART Graciela Raquel, ¹CABRERA, Sergio Alberto, ¹BENÍTEZ, Horacio Duval, ²FALAPPA, Marcelo Alejandro ²COBO, María Laura, ²MARTÍNEZ, Diego César

¹Facultad de Ciencias de la Administración - Universidad Nacional de Entre Ríos
Monseñor Tavella 1424 – Concordia, Entre Ríos (3200) - Tel.: +54(0345)4231433
{marben, anacar, caralv, migfer, graetc, sercab, duvben}@fcad.uner.edu.ar

²Departamento de Ciencias e Ingeniería de la Computación - Universidad Nacional del Sur
Avenida Alem 1253 - Bahía Blanca (B8000CPB) - Tel.: +54(0291)4595135
{mfalappa, mlc, dcm}@cs.uns.edu.ar

Resumen

Los lenguajes de programación (LP) han tenido un desarrollo creciente y se han adaptado a las necesidades de diferentes dominios de aplicación, liberando versiones periódicamente y adaptados a estándares. Este crecimiento, generó dentro de los LP, adaptaciones a pautas de diseño e implementación de programas, permitiendo su utilización en el desarrollo de aplicaciones para bases de datos (BD), entre otras.

Actualmente, una de las áreas con mayor crecimiento en el desarrollo de sistemas informáticos, es el de la programación Web. Este dominio de aplicación, ha permitido interactuar con un gran número de desarrollos tanto académicos, como comerciales, gubernamentales, industriales y para dispositivos móviles.

El presente trabajo, en base a ciertas características, parámetros, encuestas a desarrolladores y rankings de la industria; tiene como objetivo delimitar el universo de posibles lenguajes de programación orientados a objetos que posteriormente serán sometidos al estudio comparativo y análisis de rendimiento en las subsiguientes etapas previstas por el proyecto.

Palabras Claves

Lenguajes de Programación Orientados a Objetos, Lenguajes de Manipulación de Datos, Bases de Datos, Criterios de Selección de Lenguajes de Programación.

Contexto

Este trabajo se desarrolla dentro del Proyecto de Investigación y Desarrollo PID 7042 “*Estudio Comparativo y Análisis de Rendimiento de los Lenguajes de Manipulación de Datos en Bases de Datos Orientadas a Objetos y Bases de Datos Objeto-Relacionales*”, cuyo período de ejecución será desde octubre 2014 a octubre 2017, en el marco de un Acuerdo de Colaboración Académico-Científico entre la Facultad de Ciencias de la Administración de la Universidad Nacional de Entre Ríos (UNER) y el Instituto de Ciencias e Ingeniería de la Computación (ICIC) del Departamento de Ciencias e Ingeniería de la Computación (DCIC) de la Universidad Nacional del Sur (UNS).

Uno de los principales objetivos de este proyecto es la formación de recursos humanos para investigación en la Facultad de Ciencias de la Administración de la UNER, especializados en la línea prioritaria de investigación denominada “Ingeniería de Software y Lenguajes de Programación” establecida por Res. 25/11 del C.D. Al ser también ésta una de las principales líneas de investigación del Departamento de Ciencias e Ingeniería de la Computación de la UNS, se justifica la creación de un equipo de investigación inter-universidades que sea contenedor del desarrollo de investigadores de la UNER en el área.

Introducción

En la actualidad, los lenguajes de programación poseen interfaces de desarrollo de aplicaciones y acceso a sistemas de base de datos. Las aplicaciones que se desarrollan, poseen interfaces amigables para el usuario y permiten interactuar directamente con la base de datos.

Existen varias técnicas para incluir estas interacciones en un lenguaje de programación: una consiste en escribir programas de aplicación en lenguajes de alto nivel, que integran instrucciones que permiten soportar la funcionalidad de una base de datos; o bien utilizar una técnica más dinámica pero más compleja, como las API's (API: *Application Programming Interface*), que son librerías de funciones y procedimientos que pueden ser utilizados por otro software como una capa de abstracción; en el contexto de bases de datos.

Estas dos metodologías son las más comunes, pero presentan problemas derivados de las diferencias entre el modelo de la base de datos y el modelo del lenguaje de programación. Este problema disminuye con el uso de una tercera técnica, que es la utilización de un lenguaje de programación específico para construcción de aplicaciones que tienen mucha interacción con la base de datos, brindando así, compatibilidad entre ambos modelos.

Las bases de datos son masivamente utilizadas en las aplicaciones de hoy en día. Detrás de la mayoría de los sistemas informáticos, existe una base de datos así como un sistema de manejo de la misma, que permite el acceso a los datos, brindando seguridad al usuario, recuperación ante fallos, posibilidad de acceso concurrente, y mecanismos de control de concurrencia que garantizan la atomicidad de las transacciones, y la correcta modificación de los datos.

Las bases de datos y los lenguajes de programación han tenido un desarrollo creciente y, en la mayoría de los casos, se han ido adaptando a las necesidades propias de los diferentes dominios de aplicación. Los lenguajes de programación, en particular, son una herramienta de vital importancia para el desarrollo de muchas aplicaciones de software. Éstos poseen características particulares, tales como simplicidad, legibilidad, facilidad de

escritura, facilidades para la auto-documentación, herramientas para un adecuado diseño de interfaces, que procuran la producción de software con cualidades tales como confiabilidad, mantenibilidad y eficiencia, entre otras.

En la actualidad, el área con mayor crecimiento es el de la programación Web, dado que permite interactuar con un gran número de aplicaciones tanto académicas, como comerciales, gubernamentales, industriales y para dispositivos móviles.

Basándonos en esta tendencia y en las características que deberían tener los LP que se elijan para el desarrollo de este proyecto, se tendrán en cuenta encuestas a desarrolladores ("Hammer") y ciertos rankings determinados por la industria.

El sitio "The Hammer Principle"¹, ofrece resultados de encuestas anónimas realizadas a programadores donde se comparan aproximadamente cien características de cincuenta LP. Estas encuestas brindan información acerca de las propiedades de los LP tales como correctitud, simplicidad, eficiencia, entre otras.

De estas características, las relacionadas directamente con el proyecto son los siguientes: "Legibilidad (Code written in this language is very readable)", "Buena documentación (This language is well documented)", "Eficiencia (Programs written in this language tend to be efficient)", "Reusabilidad (This language encourages writing reusable code)", "Proyectos para dominios de aplicación Web (I would use this language for a web project)", "Proyectos para desarrollo de aplicaciones con interfaces amigables (I would use this language for a desktop GUI Project)".

En la industria, se encuentran disponibles rankings que permiten establecer, entre otras cosas, los LP más populares utilizados en el desarrollo de sistemas informáticos. Estos rankings o índices, se realizan a partir de información disponible en la Web y la selección se basa en los siguientes parámetros: análisis de consultas que se realizan en los

¹ "The Hammer Principle" es un sitio web donde se pueden comparar varias características de LP, basadas en distintos tipos de indicadores.

motores de búsqueda más utilizados, actividad en comunidades de programadores (proyectos, comunidades y foros), demanda de conocimientos/experiencia de programación en el mercado laboral, volumen de libros vendidos (datos de editoriales especializadas), encuestas realizadas a programadores, posibilidades tecnológicas e impacto de los mismos en el mercado; entre otras.

Tiobe

Para evaluar la popularidad de un lenguaje, Tiobe utiliza el número de veces que se ha buscado un lenguaje de programación a través de motores de búsqueda. El índice se actualiza una vez al mes. Realiza un análisis cuantitativo. En lugar de evaluar el uso real del lenguaje en toda la industria, Tiobe analiza varios motores de búsqueda (Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube y Baidu) que utilizan un algoritmo fijo.

PYPL

El ranking PYPL (Popularidad del Índice de Lenguaje de Programación) utiliza Google Trends para mostrar la frecuencia de búsqueda de tutoriales para un lenguaje específico (usando Google).

GitHut

GitHut es un ranking bastante nuevo. Analiza la cantidad de código por lenguaje de programación utilizado a través de los repositorios alojados en GitHub, donde la comunidad de desarrollo de código abierto ofrece acceso a la mayoría de sus proyectos.

Líneas de investigación y desarrollo

Existen lenguajes de programación que ofrecen soporte para acceder y manipular datos en bases de datos. El presente proyecto de investigación limitará el estudio de lenguajes de programación orientados a objetos, a aquellos que permitan la posibilidad de manipular objetos persistentes, así como lenguajes de manipulación de datos en bases de datos que, además, provean facilidades para el manejo de interfaces gráficas. Cada uno de estos lenguajes, resulta más o menos adecuado que otro para producir software con ciertas cualidades, en función de las características que ellos presentan. Para poder determinar claramente si el software que producen posee las cualidades de interés para el presente

trabajo, es necesario conocer sus características, su funcionalidad, su estructura; y efectuar mediciones, con la finalidad de comparar los resultados. En base a los lenguajes seleccionados, las cualidades del software y las características elegidas para el estudio, se podrán conocer aquellos lenguajes que experimenten un mejor comportamiento en aplicaciones orientadas a sistemas informáticos con bases de datos, para luego poder compararlos con sistemas que utilicen bases de datos objeto-relacionales.

Resultados y Objetivos

El objetivo del proyecto de investigación referenciado en el contexto es realizar un estudio comparativo y análisis de rendimiento de lenguajes orientados a objetos, así como lenguajes de manipulación de datos en bases de datos objeto-relacionales. En cada caso, se buscará medir la calidad del software producido con ellos.

En esta etapa del proyecto, y en base a determinadas características y parámetros, se delimitará el universo de posibles lenguajes de programación orientados a objetos que posteriormente serán sometidos al estudio comparativo y análisis de rendimiento.

De los indicadores obtenidos de Hammer que guardan una relación directa con las características establecidas por el proyecto para los LP, se ha diseñado la Tabla 1. En la misma, se detallan los primeros diez lenguajes que cumplen con las características analizadas.

Tabla 1:

O R D E N	Legibili- dad	Buena docu- men- tación	Eficien- cia	Reusabili- dad	Dominios de aplica- ción Web	Desarro- llo de aplicacio- nes con interfaces amiga- bles
1	Python	Java	Assem- bler	Ada	Javascript	C#
2	Eiffel	Python	C	Haskell	Ruby	Haxe
3	Haxe	Mathema- tica	Forth	Eiffel	Scala	Delphi
4	Go	C#	Fortran	O'Caml	Python	Objective C
5	Lua	Ada	C++	D	Haxe	Scala
6	F#	C	Eiffel	Common Lisp	Clojure	Java
7	Smalltalk	Common Lisp	D	Clojure	Groovy	Visual Basic
8	Ruby	Perl	Ada	Scala	PHP	F#
9	Groovy	Factor	O'Caml	F#	Erlang	Python
10	Haskell	Objective C	Go	Smalltalk	Java	Clojure

En lo que respecta a los rankings establecidos en la industria y que guardan una relación directa con el proyecto, se ha diseñado la Tabla 2, que también selecciona los primeros diez lenguajes.

Tabla 2: Rankings a febrero 2015

Orden	TioBe	GitHut	PYPL
1	C	JavaScript	Java
2	Java	Java	PHP
3	C++ (con tendencia ascendente)	Python	Python
4	Objective-C (con tendencia descendente)	CSS	C#
5	C#	PHP	C++
6	JavaScript (con tendencia ascendente)	Ruby	C
7	PHP (con tendencia descendente)	C++	Javascript
8	Python	C	Objective-C
9	Visual Basic .NET (con tendencia ascendente)	Shell	Matlab
10	Visual Basic	C#	R (con tendencia ascendente)

Las delimitaciones establecidas, proveen un universo de LP orientados a objetos que son de interés para el proyecto, para su posterior elección.

Posteriormente el proyecto de investigación contempla actividades para el logro de los siguientes objetivos:

- Realizar un estudio comparativo en base a los lenguajes seleccionados y a las características de los mismos, en función de las cualidades deseables del software que producen.

- Comparar globalmente sistemas desarrollados en lenguajes orientados a objetos que manipulen objetos persistentes con sistemas que utilicen bases de datos objeto-relacionales.

- Analizar los resultados del estudio comparativo efectuado y establecer un diagnóstico del rendimiento de los lenguajes de manipulación de datos estudiados.

Formación de recursos humanos

El equipo está integrado por un Director, un codirector, y docentes investigadores pertenecientes a la UNER y a la UNS que se desempeñan en cátedras relacionadas

directamente con el tema central de la investigación.

Se prevé que el personal docente de la UNER dedicado al proyecto inicie, avance y/o concluya sus estudios de posgrado, así como también la incorporación de becarios de investigación y la dirección de tesinas finales de grado.

El codirector del proyecto, Marcelo Gabriel BENEDETTO, es Magíster en Sistemas de Información, se encuentra realizando los cursos del Doctorado en Informática de la Universidad de Murcia (España) y cursando la Maestría en Desarrollo Local en la Universidad Nacional de General San Martín. La integrante Licenciada en Sistemas de Información Ana Lía Ramona CARABIO se encuentra realizando la Maestría en Redes de la UNLP, restando la presentación de la Tesis; el integrante Licenciado en Sistemas Sergio Alberto CABRERA se encuentra realizando la Maestría en Sistemas de Información (MSI) de la Facultad de Ciencias de la Administración de la UNER, restando sólo la presentación de la Tesis; la Licenciada en Sistemas Graciela Raquel ETCHART se encuentra realizando cursos válidos para la obtención de créditos del Magister en Ciencias de la Computación de la UNS. Finalmente, el Licenciado en Sistemas Duval Horacio BENITEZ se encuentra inscripto para la nueva cohorte de la MSI de la Facultad de Ciencias de la Administración de la UNER, donde realizará su trabajo de investigación en el área del proyecto.

Referencias

ELMASRI, Ramez; NAVATHE, Shamkant B. Fundamentals of Database Systems (Sixth Edition). Addison Wesley, 2010.

Índice GitHut en <http://github.info/>. Recuperado el 21 de febrero de 2015.

Índice PYPL en <http://pypl.github.io/PYPL.html>. Recuperado el 21 de febrero de 2015.

Índice TIOBE, en <http://www.tiobe.com/index.php/content/paper/info/tpci/index.html>. Recuperado el 21 de febrero de 2015.

LOUDEN, Kenneth C. Lenguajes de Programación: Principios y Práctica. Segunda Edición. México, Thomson Internacional, 2004.

MACIVER, David R. "The Hammer Principle", en <http://hammerprinciple.com/therighttool>. Recuperado el 21 de febrero de 2015.

MEYER, Bertrand. Object-Oriented Software Construction. Second Edition. 17th Printing 2011. United States of America, Prentice-Hall, 1997.

MEYEROVICH, Leo A.; RABKIN, Ariel S. Empirical Analysis of Programming Language Adoption. En ACM SIGPLAN Notices. ACM, 2013. p. 1-18.

MEYEROVICH, Leo A.; RABKIN, Ariel S. Socio-PLT: Principles for Programming Language Adoption. En Proceedings of the ACM International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software. ACM, 2012. p. 39-54.

PRATT, Terrence W.; ZELKOWITZ, Marvin V. Programming Languages. Design and Implementation. Fourth Edition. United States of America, Prentice-Hall, 2001.

Programming Language Popularity, en <http://www.langpop.com/>. Recuperado el 21 de febrero de 2015.

SEBESTA, Robert W. Concepts of Programming Languages. Tenth Edition. Addison-Wesley, 2012.

SILBERSCHATZ, Abraham; KORTH, HENRY; SUDARSHAN, S. Database System Concepts. Sixth Edition. McGraw-Hill Science/Engineering/Math, 2010.