

# Añadiendo Capacidades de Búsqueda a Entornos de Desarrollo Integrado para la Enseñanza y Aprendizaje de la Programación en Ambientes Remotos

Adding Search Features to Integrated Development Environments for Teaching and Learning Programming in Remote Scenarios

Rolando Salazar-Hernández<sup>1</sup>, Clarisa Pérez-Jasso<sup>1</sup>, Julio Rodríguez-Cano<sup>2</sup>, Eduardo Pérez-Perdomo<sup>2</sup>

<sup>1</sup> Universidad Autónoma de Tamaulipas, México

<sup>2</sup> Universidad de Holguín, Cuba

rsalazar@uat.edu.mx , clperez@uat.edu.mx , jcrodriguez@facinf.uho.edu.cu ,  
eperezp@facinf.uho.edu.cu

**RESUMEN.** La comunidad de varios Entornos de Desarrollo Integrado (IDEs) ha mostrado interés en la construcción de herramientas con soporte para la colaboración de los desarrolladores, pero sin hacer hincapié en la búsqueda de código fuente. En este trabajo se presenta un plug-in para los IDEs NetBeans y Eclipse en el contexto académico como una alternativa a esta disyuntiva, con el cual se realizó una experimentación con dos grupos de estudiantes que arrojó como conclusión que con la aplicación de técnicas de búsqueda en los IDEs, se pueden obtener mejores resultados respecto a herramientas de búsqueda de código fuente centradas en una interacción individual y los IDEs por separado.

**ABSTRACT.** The community of many Integrated Development Environments (IDEs) has shown interest to develop tools with support for the collaboration among developers, but without doing emphasis in source code search mechanisms. In this work we propose a NetBeans and Eclipse plug-in for the academic context as an alternative to the problem mentioned above. Also, we carry out an experimentation with two groups of students, that showed as main conclusion that with the integration of search techniques inside IDEs we can obtain better results in relation to use search tools and IDEs separately.

**PALABRAS CLAVE:** Búsqueda de Código Fuente, Aprendizaje Colaborativo, Interfaz Multiusuario.

**KEYWORDS:** Source Code Search, Collaborative Learning, Multiuser Interface.

## 1. Introducción

La programación de aplicaciones informáticas o desarrollo de software es una actividad que suele ser compleja de efectuar, aun cuando hoy contamos con gran diversidad de lenguajes y bibliotecas de clases que evitan en cierta medida que se deba partir desde cero para la implementación de nuevas soluciones computacionales. Es por eso que los desarrolladores de software pasan buena parte del tiempo buscando información técnica, al enfrentarse a numerosas necesidades que emergen a partir de tareas diarias relacionadas con la programación, tales como escribir y reutilizar código fuente, arreglar errores, encontrar documentación y ejemplos para el aprendizaje de una nueva tecnología, probar nuevos algoritmos, explicar fragmentos de código a un grupo de colegas fuera o dentro de un proyecto, entre otros. Este mismo dinamismo es más acuciante en el proceso de enseñanza-aprendizaje de la Programación en las escuelas que forman estudiantes con perfiles relacionados con la Informática o Ciencias de la Computación, entre otras disciplinas según el país.

En los últimos años ha habido un gran esfuerzo por parte de investigadores de diferentes campos académicos y empresariales hacia la construcción de herramientas que permitan al desarrollador realizar búsquedas, y otras actividades referentes al código, de forma que satisfagan sus necesidades de información. Ejemplo de ello son los motores de búsqueda tales como, Google Code Search, Koders, Krugle, Codefetch y Codase. La actividad de búsqueda es considerada una herramienta de gran importancia en el proceso de desarrollo de software [Bajracharya, Kuhn y Ye, 2009). Los investigadores Ken Krugler y John D. Mitchell (2007) enfatizan que: "Alrededor del 25% del tiempo de trabajo de un desarrollador es ocupado en búsquedas de información. Aunque está bien empleado, porque encontrar código reutilizable puede propiciar que un proyecto esté terminado en tiempo y con resultados de alta calidad". Sin embargo, ninguna de las soluciones identificadas en este estudio posibilitan la colaboración entre los estudiantes y los profesores durante realización de tareas de programación.

Así, la estructura de este trabajo es la siguiente. En la Sección 2 hacemos una descripción de los principales elementos que aporta la solución propuesta al proceso de enseñanza-aprendizaje. Luego, en la Sección 3 se incluye una evaluación donde se compara el empleo de herramientas tradicionales con nuestra solución para resolver una tarea de programación en Java. Finalmente, se presentan las conclusiones en la Sección 4.

## 2. Descripción del plug-in propuesto

Al igual que en la industria del desarrollo de software, las universidades y otros centros educacionales que enseñan programación, utilizan como herramienta principal a los IDEs. En estas herramientas la búsqueda es un medio útil de navegación. Varios investigadores se han referido a la búsqueda que se encuentra embebida en las funcionalidades de navegación de un IDE como navegación guiada por búsquedas (Kuhn, 2010). En nuestro caso, la meta de esta actividad es que los estudiantes y profesores ahorrasen tiempo en la comunicación, análisis, discusión y desarrollo de ejercicios de programación.



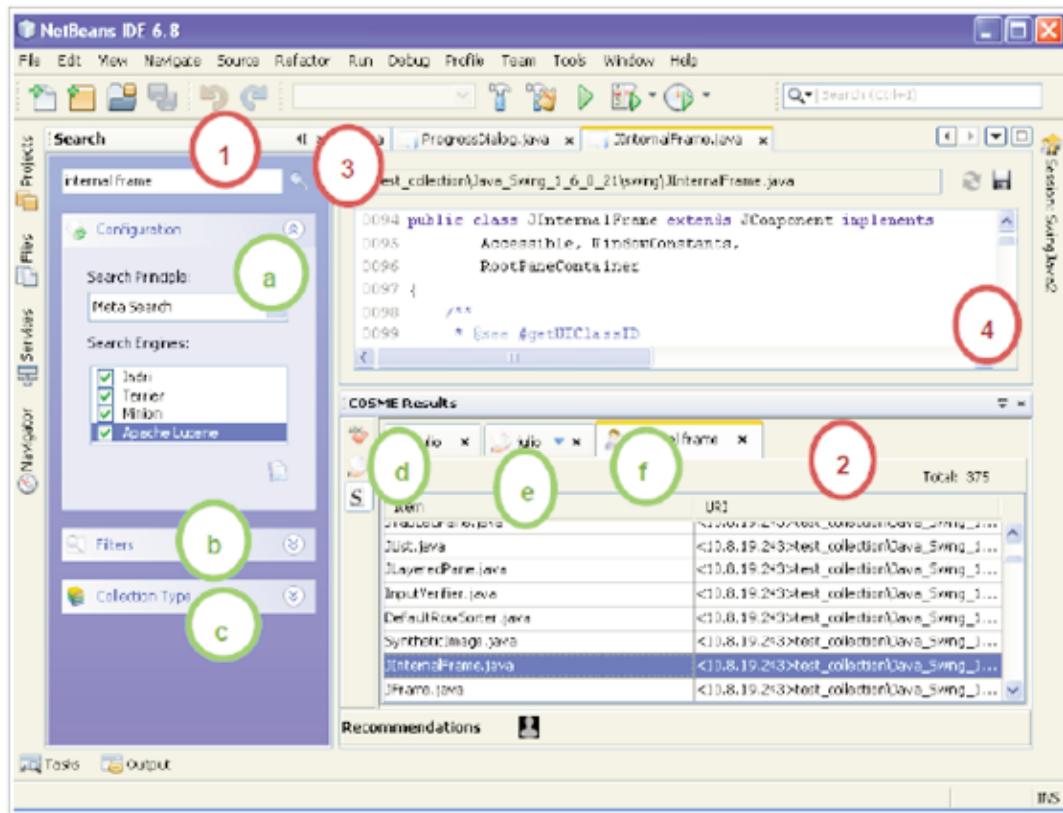


Figura 1. Interface multiusuario del IDE NetBeans con el plug-in instalado.

La Figura 1 muestra una búsqueda realizada desde el IDE NetBeans con el plug-in propuesto instalado. Se encuentran desplegadas tres ventanas, las cuales son denominadas TopComponents según el API Window System de la plataforma NetBeans: (1) Ventana de búsqueda, está integrada a su vez por tres paneles plegables: (a) Configuration: donde el desarrollador selecciona el principio de búsqueda y el motor por el cual desea realizar la búsqueda; (b) Filters: muestra las partes de una clase a la cual pertenece el código que se necesita encontrar y (c) Collection Type: permite especificar la fuente documental por la cual el desarrollador solicita los resultados, libros (pdf, txt) o código Java. (2) Ventana de resultados de búsqueda, los resultados pueden clasificarse de acuerdo a tres orígenes diferentes: (f) búsqueda local del desarrollador (dentro o fuera de su sesión), (d) búsqueda del jefe de la sesión colaborativa (generalmente el profesor) o pueden ser resultados recomendados explícitamente (e) por otro desarrollador en su propia sesión de búsqueda colaborativa. (3) Ventana que visualiza los resultados de búsqueda, muestra resultados en formato pdf, txt o Java. También está activa la ventana que muestra la sesión de búsqueda a la cual entró el usuario, en este caso llamada "SwingJava2" que está colapsada (4). Una representación similar tendríamos para el IDE Eclipse.

### 3. Evaluación

Para conocer la efectividad del plug-in propuesto, se llevó a cabo un experimento, el cual fue realizado con estudiantes y profesores de la Universidad de Holguín, para buscar código fuente desde dos grupos: uno para la búsqueda con herramientas tradicionales (G1) y el otro utilizando nuestra propuesta (G2). Para la evaluación se emplearon las métricas propuestas por Pickens et al. (2008). Para realizar el análisis estadístico se utilizó la prueba no paramétrica de Wilcoxon, y se empleó el método de Monte Carlo, con intervalos de confianza del 99% y un número de muestras igual a 10 000. Quedando definida como hipótesis nula (H0) que entre G1 y G2 no existen diferencias significativas ( $MG1 = MG2$ ), y como hipótesis alternativa (H1), la existencia de diferencias significativas entre G1 y G2 ( $MG1 \neq MG2$ ). Como conclusión de este experimento se pudo comprobar

que  $MG1 < MG2$  con diferencias significativas, para la Precisión y Exhaustividad de Selección. Por tanto se rechaza  $H_0$  y se acepta  $H_1$ .

#### 4. Conclusiones

La búsqueda de código fuente es un área de investigación reciente que responde al problema de satisfacer las necesidades de apoyar la enseñanza-aprendizaje de la programación. Como parte de experimentos y estudios realizados varias herramientas integradas a IDEs y fuera de éstos se han construido. Sin embargo, pocos han sido los intentos de unificar la búsqueda y la colaboración como actividades fundamentales del estudio de la programación por grupos de estudiantes. La flexibilidad y escalabilidad de los IDEs de código abierto como el NetBeans y el Eclipse permitieron la construcción de un plug-in que permite la colaboración entre estudiantes y profesores durante la búsqueda de código fuente en entornos de redes distribuidas, y de esta forma contar con mejores alternativas a las herramientas tradicionales de búsqueda y desarrollo de software que se encuentran separadas.

#### Agradecimientos

Agradecemos a la Universidad Autónoma de Tamaulipas por brindar apoyo a la realización del proyecto. En especial al director de U.A.M.Mante.

Cómo citar este artículo / How to cite this paper

Salazar-Hernández, R., Pérez-Jasso, C., Rodríguez-Cano, J., y Pérez-Perdomo, E. (2015). Añadiendo Capacidades de Búsqueda a Entornos de Desarrollo Integrado para la Enseñanza y Aprendizaje de la Programación en Ambientes Remotos. *Campus Virtuales*, Vol. 4, Num. 2, pp. 66-69. Consultado el [dd/mm/aaaa] en [www.revistacampusvirtuales.es](http://www.revistacampusvirtuales.es)

#### Referencias

- Bajracharya, S., A. Kuhn, y Y. Ye. (2009). Sourcerer: An Internet-Scale Software Repository in SUITE 2009: First International Workshop on Search-Driven Development– Users, Infrastructure, Tools and Evaluation 2009. Vancouver, Canada.
- Krugler, K. y Mitchell, J.D. (2007). Search-driven development: Five reasons why search is your most powerful tool. Disponible en: <http://www.linuxworld.com/news/2007/012907-search.html>
- Kuhn, A. (2010). Immediate Search in the IDE as an Example of Socio-Technical Congruence in Search-Driven Development. In SUITE '10 May 2010. Cape Town, South Africa.
- Pickens, J., et al. (2008). Algorithmic Mediation for Collaborative Exploratory Search. In SIGIR'08. 2008. Singapore.

