# Estado del Arte Para Herramientas de Comprensión de Software

A continuación se presentan los hallazgos correspondientes a herramientas que permiten entender la estructura de una aplicación en cualquiera de sus niveles a través del código fuente. Cada una se clasifica de la siguiente manera.

|  |  |
| --- | --- |
| **Clasificación de Hallazgos** | **Significado.** |
|  | Herramientas dedicadas a la comprensión de software que permiten obtener vistas y operaciones de alto y medio nivel (grupos funcionales, arbol de llamados, subsistemas,etc). Pueden incluir vistas de bajo nivel. |
|  | Herramientas ligeras de comprensión de software con vistas y operaciones de bajo nivel, y métricas de código. También incluye herramientas de comprensión especializadas. |
|  | Entornos de desarrollo de software que incluyen potentes herramientas de comprensión de software (tanto operaciones como vistas de alto y medio nivel). |
|  | Software de comprensión que permite realizar ''add-on's'' personalizados, aumentando la cantidad de vistas y operaciones disponibles en el mismo. |

*Tabla 1: Clasificación de Herramientas.*

La siguiente tabla contiene la información encontrada con la anterior nomenclatura.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Herramienta.** | **Motivación.** | **Dominio.** | **Lenguaje.** | **Operaciones.** | **Vistas.** | **Fuente.** |
| Moose | Herramienta que permite a los programadores crear análisis personalizados de manera rápida. | Entendimiento de Programas legado Java. | Java. | * Personalizados (custom analisis). | * Neural. * Jerárquica. * MetaBuscadores. * Query Browsers. * TreeMaps. * Personalizdos. | http://www.moosetechnology.org/#features |
| Vizz Analizer | Herramienta en desarrollo que flexibilice la forma de hacer análisis de código, y a su vez ofrecer vistas fáciles de entender. | Programación Orientada a Objetos de Java. | Java. | * Análisis de bajo, medio y alto nivel. * Personalizados a través de extensiones. | * Xml. * AST. * Grafo de Cercanía de Tablas. | http://arisa.se/files/LELPP-03.pdf |
| Extension de EAR. | Extensión que quiere mejorar el entendimiento de algoritmos de enrutamiento sobre la herramienta EAR. | Programas con algoritmos de enrutamiento complejos. | Todos (es necesario parser). | * Inspección de código por anotaciones y algoritmo de enrutamiento. | * Árbol de llamado entre funciones. * Grafo de dependencia entre módulos. * Outline de funciones. * Mapa de Algoritmo de Enrutamiento. | https://bibliotecadigital.ipb.pt/bitstream/10198/584/1/CACIC06.pdf |
| Concern Highlight | Herramienta que nace para entender particularidades de la programación orientada a aspectos que son difíciles de entender en el código. | Programas Orientados a Aspectos. | Java (Plugin Eclipse) | * Analizador de "Concerns" en programas. | * Visualizador de código por concerns. | http://www.ics.uci.edu/~andre/papers/C53.pdf |
| Imagix 4D | Herramienta que facilita la comprensión de código legado complejo en C, C++ y Java. | Software legado. | C, C++, Java. | * Analizador de Dependencias. * Analizador de Flujo de Datos. | * Diagrama de Subsistemas. * Clases UML. * Estructuras De matrices de diseño. * Herencia de clases. * Vista de Niveles de Abstracción. | http://www.imagix.com/products/source-code-analysis.html |
| Sniff + | IDE de desarrollo que permite entender visualmente el código realizado en la misma. | Desarrollo de Software legado. | C, C++, Java, IDL, Fortran 77. | * Análisis de referencias cruzadas. * Análisis de dependencias. * Análisis de herencia de clases. | * Vista de referencias Cruzadas. * Vista de imports. * Vista jerárquica. | http://www.linuxfocus.org/English/March2000/article140.shtml |
| Source Insight. | IDE de desarrollo para java , C, C++ y C# que facilita el entendimiento del código a medida que el mismo se programa. | Entendimiento de código fuente en desarrollo Orientado a Objetos. | C, C++, C# y Java. | * Análisis de referencias cruzadas. * Análisis de herencia de clases. * Análisis estructural de código. | * Arboles de Invocación de llamadas. * Diagramas de Clase. * Herencia de clases. | http://sourceinsight.com/ |
| Understand. | IDE de desarrollo que ofrece métricas y vistas para entender código que crece con el tiempo facilmente. | Entendimiento de código en el desarrollo. | COBOL, Ada, C, C++, C#, Java, Phyton, Fortran, Web, Entre otros. | * Análisis de Referencias Cruzadas. * Análisis Estructural. * Análisis de calidad. * Análisis personalizados. | * Diagramas de clases. * TreeMaps. * Árbol de Flujo de Control. * Arboles de jerarquía de llamados. * Grafo de Dependencias. * Grafo de Declaraciones. | https://scitools.com/ |
| CBrowser + CScope. | Herramienta de análisis, indización y búsqueda en código legado de fácil instalación. | Código procedimental C, C++. | C, C++. | * Análisis de referencia cruzadas. * Análisis jerárquico de llamadas. | * Arboles de Extensión * Grafo de referencias cruzadas. * Outline de clases. * Documentación de Código en HTML. * Vista de Comentarios. | http://www.brothersoft.com/cc-rider-6611.html |
| CC Rider | Herramienta de visualización y generación de documentación. | Visualización estructural de código. | C, C++. | * Análisis jerárquico de llamadas. | * Árbol de llamados. * Métricas de Programa (número de líneas de código, numero de procedimientos). | http://cscope.sourceforge.net/ |
| Code Surfer | Herramienta para entender código fuente construido en C, C++ para Intel x86. Permite obtener una representación intermedia del mismo que facilita la visibilidad de la estructura semántica y comportamiento del software. | Visualizador estructural de código de nivel medio. | C, C++ | * Análisis de dependencias. * Analizador de punteros. * Inspector de directorios. * Personalizados. | * Árbol de dependencias. * Árbol de llamadas. * Búsqueda de funciones. * Personalizados. | http://www.grammatech.com/research/technologies/codesurfer |

*Tabla 2:Descripción de herramientas de comprensión.*