Tarefa 2. Procura de analizadores dinámicos de código no mercado e comparativas.

Types of dynamic analysis

Code coverage

Computing the code coverage according to a test suite or a workload is a standard dynamic analysis technique.

- Gcov is the GNU source code coverage program.
- VB Watch injects dynamic analysis code into Visual Basic programs to monitor code coverage, call stack, execution trace, instantiated objects and variables.

Memory error detection

- Intel Inspector: Dynamic memory error debugger for C, C++, and Fortran applications that run on Windows* and Linux*..
- AddressSanitizer: Memory error detection for Linux, macOS, Windows, and more. Part of LLVM.
- BoundsChecker: Memory error detection for Windows based applications. Part of Micro Focus DevPartner.
- Dmalloc, library for checking memory allocation and leaks. Software must be recompiled, and all files must include the special C header file dmalloc.h.
- Purify: mainly memory corruption detection and memory leak detection.
- Valgrind runs programs on a virtual processor and can detect memory errors (e.g., misuse of malloc and free) and race conditions in multithread programs.

Fault localization

Fault localization refers to locating the buggy code (for example the buggy statement) according to failing and passing test cases. For example, Tarantula is a well-known fault localization approach based on the covered code. Fault localization illustrates an important property of dynamic analysis: the results on the analysis depend on the considered workload, inputs or test cases. For fault localization, it has been shown that one can refactor the test cases in order to get better results.

Invariant inference

Daikon is an implementation of dynamic invariant detection. Daikon runs a program, observes the values that the program computes, and then reports properties that were true over the observed executions, and thus likely true over all executions.

Security analysis

Dynamic analysis can be used to detect security problems.

- IBM Rational AppScan is a suite of application security solutions targeted for different stages of the development lifecycle. The suite includes two main dynamic analysis products - IBM Rational AppScan Standard Edition, and IBM Rational AppScan Enterprise Edition. In addition, the suite includes IBM Rational AppScan Source Edition - a static analysis tool.

Concurrency errors

- Parasoft Jtest uses runtime error detection to expose defects such as race conditions, exceptions, resource & memory leaks, and security attack vulnerabilities.

- Intel Inspector performs run time threading and memory error analysis in Windows.
- Parasoft Insure++ is runtime memory analysis and error detection tool. Its Inuse component provides a graphical view of memory allocations over time, with specific visibility into overall heap usage, block allocations, possible outstanding leaks, etc.
- Google's Thread Sanitizer is a data race detection tool. It instruments LLVM IR to capture racy memory accesses.

Program slicing

Main article: Program slicing

For a given subset of a program's behavior, program slicing consists of reducing the program to the minimum form that still produces the selected behavior. The reduced program is called a "slice" and is a faithful representation of the original program within the domain of the specified behavior subset. Generally, finding a slice is an unsolvable problem, but by specifying the target behavior subset by the values of a set of variables, it is possible to obtain approximate slices using a data-flow algorithm. These slices are usually used by developers during debugging to locate the source of errors.

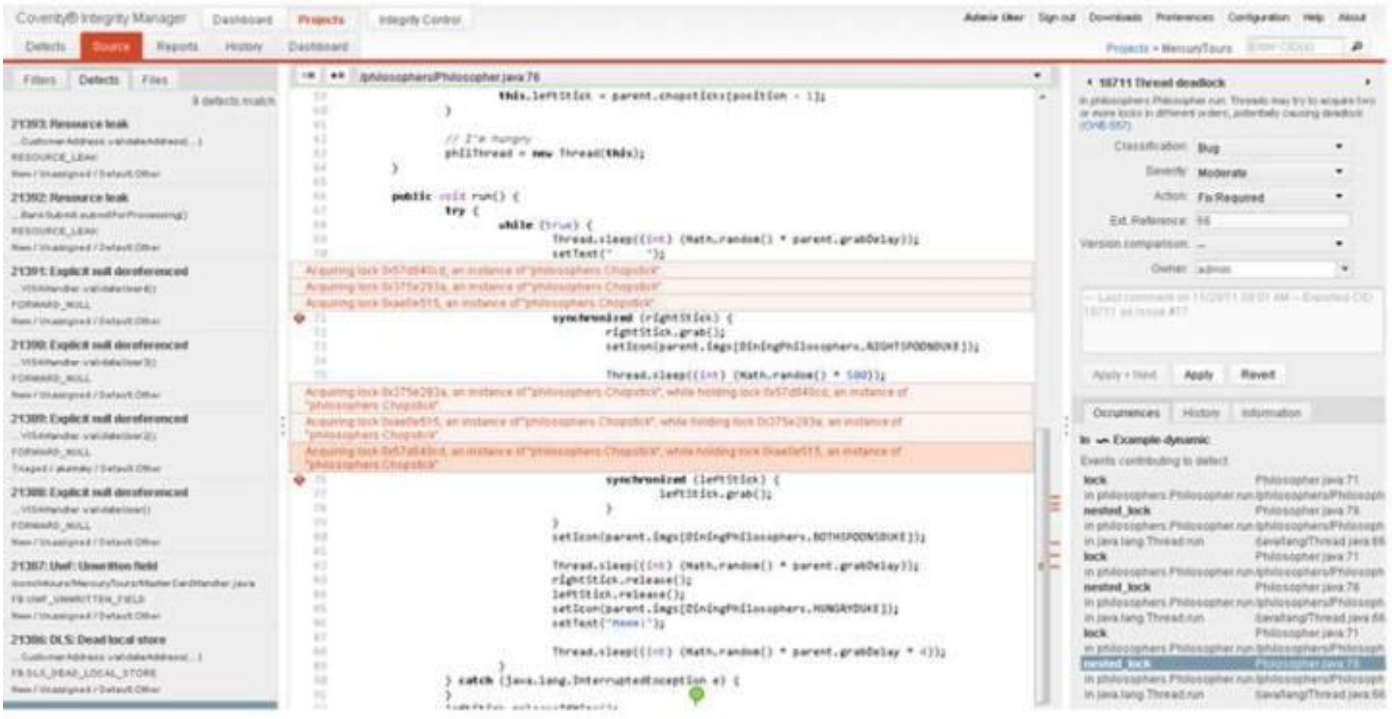
Performance analysis

Most performance analysis tools use dynamic program analysis techniques

- Prism from CriticalBlue is a tool that dynamically traces software applications at runtime and captures data that can be used to analyze and identify the causes of poor performance.

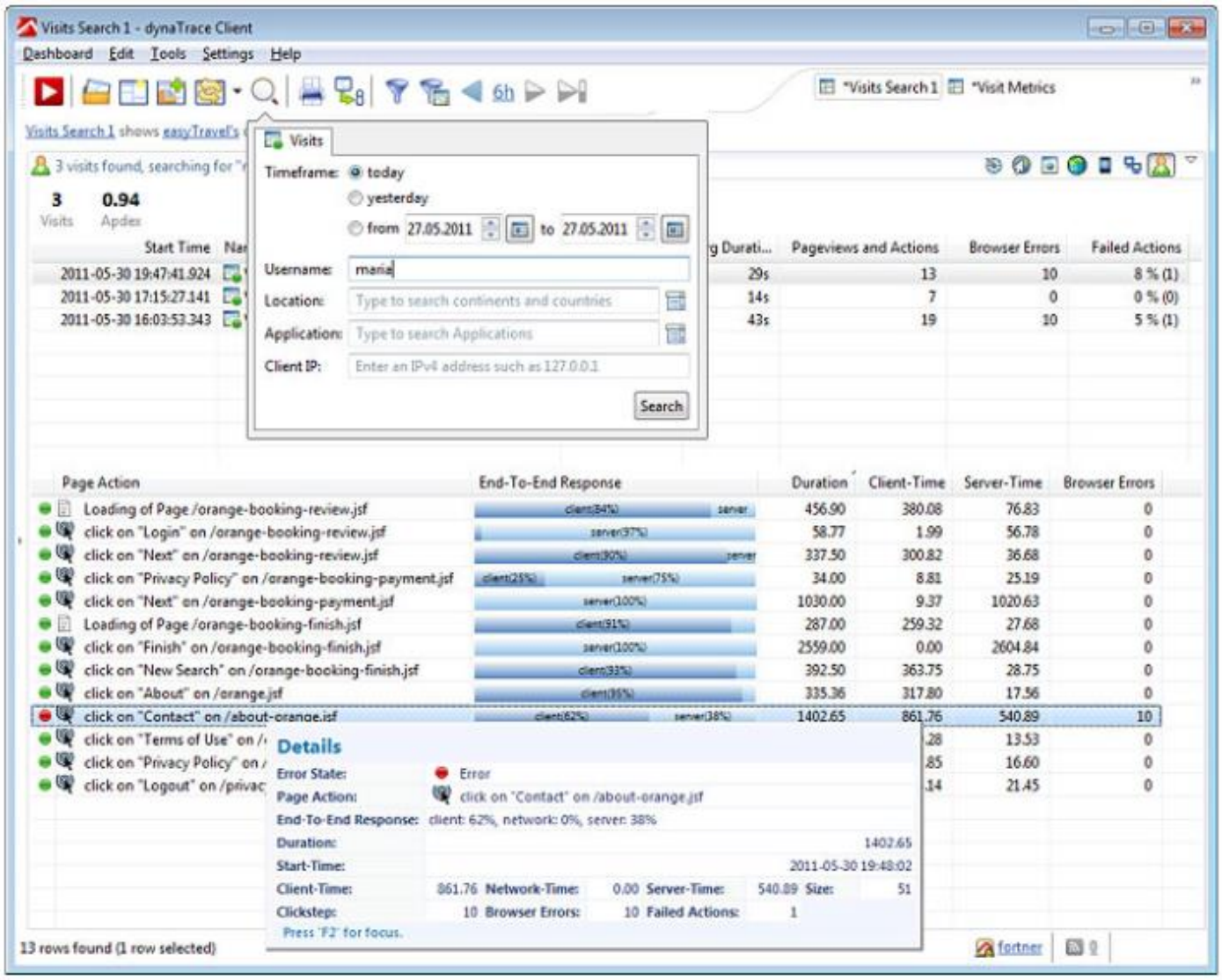
Coverity Dynamic Analysis:

· Ayuda a diagnosticar problemas instrumentando el código Java.



Compuware Dynatrace:

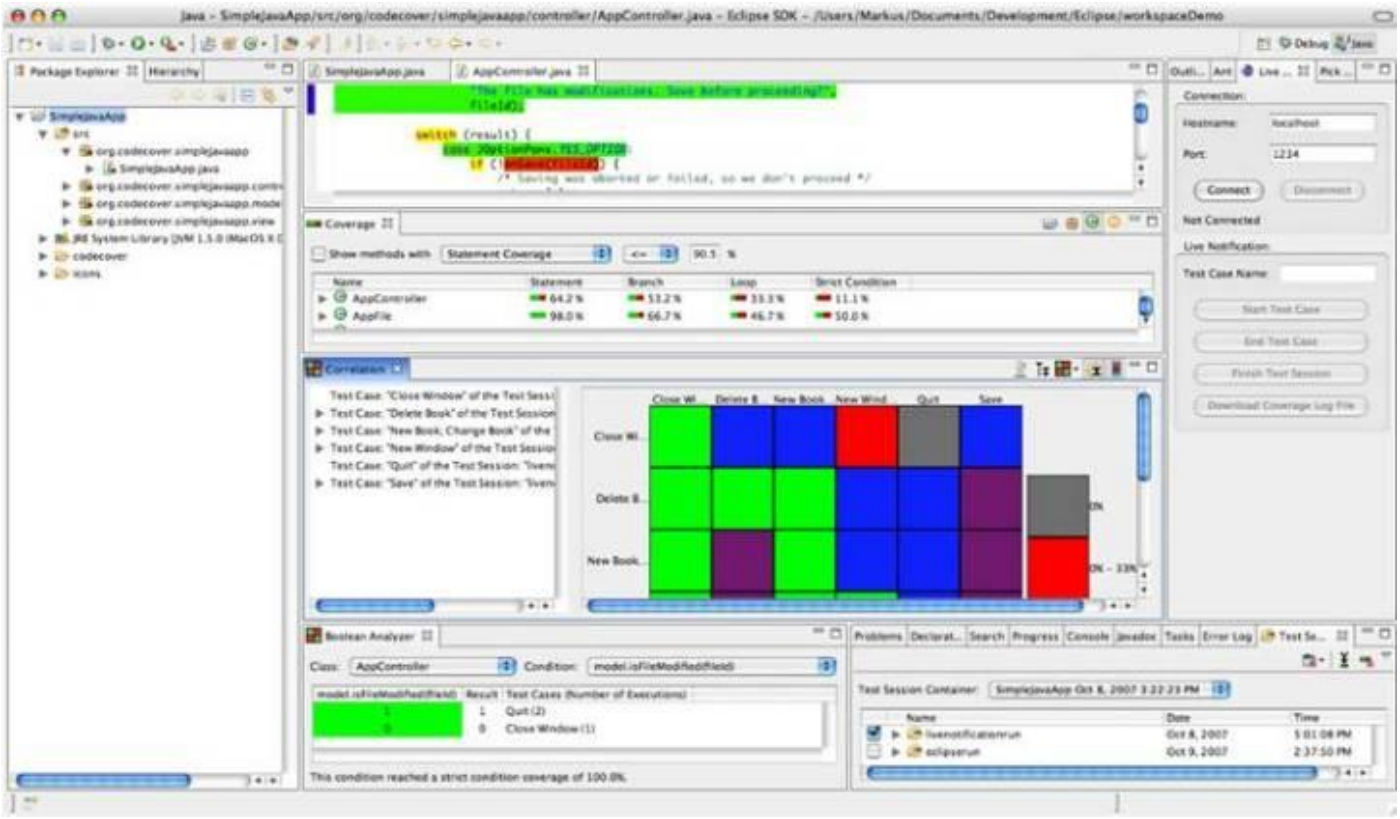
· Permite monitorizar la aplicación en producción pero también ayudarnos a detectar problemas en nuestra aplicación de calidad y rendimiento



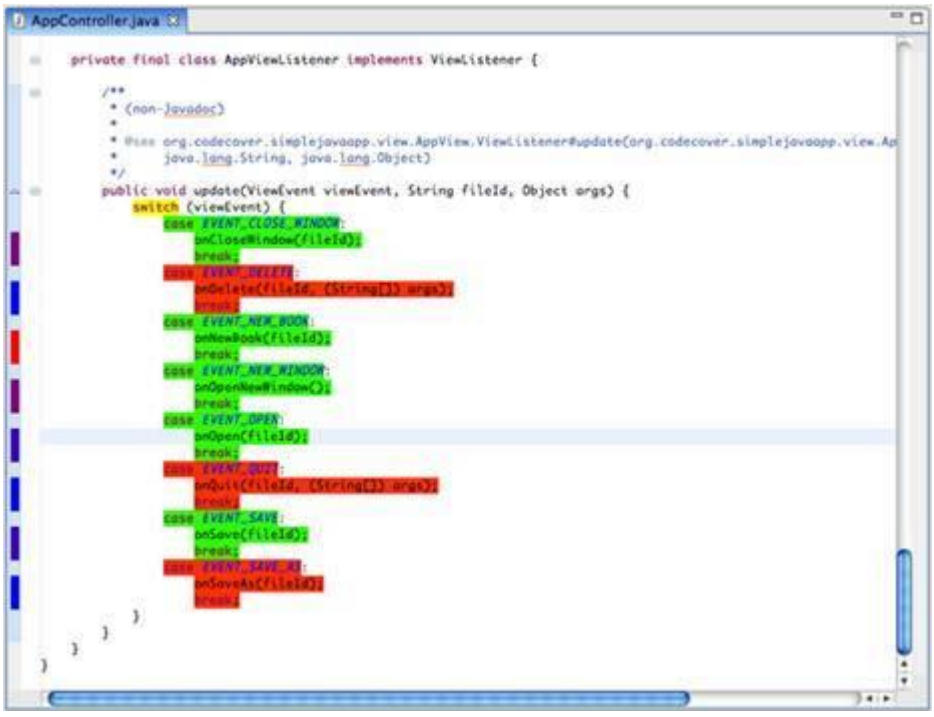
CodeCover

· Permite medir la cobertura de nuestros Tests instrumentando el código

· Ofrece un plugin de Eclipse:



Que nos permite detectar que parte del código no se prueba haciendo mutaciones sobre nuestro código:



[BACTERIO Mutation Test System:]

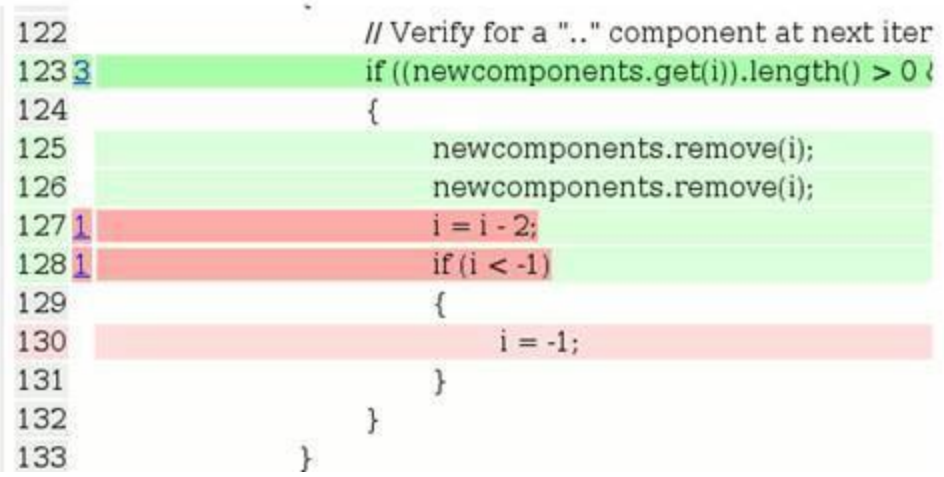
· es una completa herramienta de análisis dinámico para Java basada en la técnica de mutación

· La mutación es un técnica de pruebas que se basa en la capacidad de los casos de prueba para encontrar fallos en el sistema que se está probando.

[PITest]:

· Esta herramienta permite crear mutaciones de nuestros tests para comprobar la efectividad de nuestros tests.

· Genera informes HTML que combinan cobertura, falta de cobertura y cobertura mutaciones



jChord:

· Permite hacer análisis estático y dinámico