

Taller: Unit Testing y Cubrimiento

Parte 1: JUnit y Cobertura en Java

- 1) Instalar el plugin de eclipse para medir cobertura EclEmma (<http://www.eclemma.org/>)
- 2) Escribir un test suite para la clase `org.autotest.StackAr` que alcance 100% de cobertura de statements y de branches.

Parte 2: Una Herramienta de Cobertura para Python

- 1) Ejecutar los tests de unidad del módulo `test_cgi_decode.py` para el módulo `cgi_decode.py` ¿Cuántos tests de unidad pasan y cuántos no pasan?
Rta:
- 2) Crear un nuevo módulo nuevo que construya el AST para el módulo `cgi_decode.py` y lo imprima por consola usando el modulo "ast" de python.
- 3) Crear una clase `LineCounter` que cuente la cantidad de líneas de código (statements) hay en el módulo `cgi_decode.py`. ¿Cuántas líneas encontró?
Rta:
- 4) Crear una clase `BranchCounter` que cuente la cantidad de branches hay en el módulo `cgi_decode.py`. ¿Cuántas branches encontró?
Rta:
- 5) Completar la clase `LineCoverage` para que al ejecutarse los tests de unidad en `test_cgi_decode.py` reporte el recubrimiento de statements sobre `cgi_decode` ¿Cuál es el coverage de statements que reporta?
Rta:
- 6) Completar la clase `BrachCoverage` para que al ejecutarse los tests de unidad en `test_cgi_decode.py` reporte el recubrimiento de statements sobre el módulo `cgi_decode.py`. ¿Cuál es el coverage de statements que reporta?
Rta: