Continuos Integration Java

Estándar de código en Java

- El más utilizado es el propuesto por Sun Microsystems. Pero existen otros propuestos por: AmbySoft, BSSC, Intalio Inc., Brendon Wilson, ChiMu Corporation

Referencias

- http://www.sourceformat.com/coding-standard-java.htm
- http://www.oracle.com/technetwork/java/javase/documentation/codeconventions-139411.html#16712
- http://docs.oracle.com/javase/specs/jls/se8/jls8.pdf

Check coding standard

- Existen varias herramientas para checar el estándar de código.

https://en.wikipedia.org/wiki/List_of_tools_for_static_code_analysis#Java

Checkstyle

- http://checkstyle.sourceforge.net/
- Tiene plugins para editores como Eclipse o NetBeans que son de los más utilizados por desarrolladores Java. Y también plugin para Jenkins si se utiliza un servidor CI.

Pretty Print

- La librería jpplib es un paquete de Java para estructurar bien nuestro código. Usando saltos de línea, identación o el máximo de nuestra línea de código.

  - http://jpplib.sourceforge.net/

Doc test

- JDocTest es una implementación de Python doctest para Java. No remplaza el estándar de javadoc, se agrega al etiqueta @doc.test y tiene soporte para JUnit

- https://github.com/cscott/JDoctest

Pruebas unitarias

- Framework para escribir pruebas repetibles en Java
- http://junit.org/
- Un ejemplo de manejar Junit
- https://github.com/junit-team/junit/wiki/Getting-started

Code coverage

- Algunas herramientas para medir la cobertura de pruebas en Java:

1. Atlassian Clover
2. Cobertura
3. JaCoCo
4. Code Cover
5. PITest

- Utilizaría JaCoCo es gratuita, se integra con JUnit y Jenkins
  - http://www.eclemma.org/jacoco/

Referencia:

https://confluence.atlassian.com/display/CLOVER/Comparison+of+code+coverage+tools

Servidor continuos integration

- Jenkins

- https://jenkins-ci.org/