**Entornos de usuario**

UD5: Diseño y realización de pruebas

**Actividad 8: Ejercicios de Junit, Caja negra y Caja blanca**

**Ejercicio 1**

**Buscar herramientas alternativas a JUnit, tanto herramientas, frameworks o librerías. Pon al menos 2 alternativas y explicalas con detalle.**

**-DbUnit** es una extensión de JUnit (también utilizable con Ant) dirigidos a proyectos de bases de datos que, entre otras cosas, pone su base de datos en un estado conocido entre las corridas de prueba. Esta es una excelente manera de evitar la miríada de problemas que pueden ocurrir cuando un caso de prueba corrompe la base de datos y hace que las pruebas posteriores a fallar o exacerbar el daño.

DbUnit tiene la capacidad de exportar e importar los datos de bases de datos hacia y desde los conjuntos de datos XML. Desde la versión 2.0, DbUnit también puede trabajar con grandes conjuntos de datos cuando se utiliza en el modo de transmisión. DbUnit también le puede ayudar a verificar que los datos de base de datos coinciden con un conjunto esperado de valores.

**Selenium** es un entorno de pruebas de software para aplicaciones basadas en la web. Selenium provee una herramienta de grabar/reproducir para crear pruebas sin usar un lenguaje de scripting para pruebas (**Selenium IDE**). Incluye también un lenguaje específico de dominio para pruebas (**Selanese**) para escribir pruebas en un amplio número de lenguajes de programación populares incluyendo Java, C#, Ruby, Groovy, Perl, Php y Python. Las pruebas pueden ejecutarse entonces usando la mayoría de los navegadores web modernos en diferentes sistemas operativos como Windows, Linux y OSX.

**Ejercicio 2**

**¿Qué son los test de caja negra? Contesta a la pregunta y busca al menos 2 herramienta que sirva para realizar estos tipos test y explícalas.**

**-(Black-Box Testing)** son pruebas funcionales. Se parte de los requisitos funcionales, a muy alto nivel, para diseñar pruebas que se aplican sobre el sistema sin necesidad de conocer como está construido por dentro (Caja negra). Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si la función se está desempeñando correctamente por el sistema bajo prueba. Las herramientas básicas son observar la funcionalidad y contrastar con la especificación.

**EJEMPLOS:**

**Sonar** Herramienta de gestión de la calidad del código fuente. Permite recopilar, analizar, y visualizar métricas del código fuente. Está formado por PMD, Checkstyl, Findbugs, Clover y Cobertura. Principalmente es usado con Java pero da soporte a otros lenguajes. (LGPL)

**CodePro Analytix** También es una herramienta de gestión de la calidad del software. Ofrece un entorno para evaluación de código, métricas, análisis de dependencias, cobertura de código, generación de Test unitarios, etc

**Ejercicio 3**

**¿Qué son los test de caja blanca? Contesta a la pregunta y busca al menos 2 herramientas que sirvan para realizar estos tipos test y explícalas.**

**(White-Box Testing).** Son pruebas estructurales. Conociendo el código y siguiendo su estructura lógica, se pueden diseñar pruebas destinadas a comprobar que el código hace correctamente lo que el diseño de bajo nivel indica y otras que demuestren que no se comporta adecuadamente ante determinadas situaciones. Ejemplos típicos de ello son las pruebas unitarias. Se centran en lo que hay codificado o diseñado a bajo nivel por lo que no es necesario conocer la especificación de requisitos, que por otra parte será difícil de relacionar con partes diseñadas a muy bajo nivel.

**EJEMPLOS:**

**Selenium.** Compuesto por dos herramientas: Selenium IDE y SeleniumWebDriver. La primera, un plugin de Firefox que te genera un entorno de desarrollo y que permite crear casos de prueba para aplicaciones web. La segunda, Selenium WebDriver, ejecuta las pruebas. Este entorno de automatización de pruebas automáticas opera en los principales navegadores (IE, Mozilla, Chrome y Opera). Además, permite pruebas para dispositivos móviles, para iPhone y Android. Utiliza los siguientes lenguajes: Python, Ruby, Java y C#. La licencia es “Apache 2.0 License”.

**JMeter** es un proyecto de Apache que puede ser utilizado como una herramienta de prueba de carga para analizar y medir el desempeño de una variedad de servicios, con énfasis en aplicaciones web.

JMeter puede ser usado como una herramienta de pruebas unitarias para conexiones de bases de datos con JDBC, FTP, LDAP, Servicios web, JMS, HTTP y conexiones TCP genéricas