# IMPORTANCIA DE LAS PRUEBAS DE INTEGRACIÓN EN EL CONTROL DE LA CALIDAD DE LOS PRODUCTOS

**Autora:** Danays Rodríguez Suárez

**RESUMEN**
Las pruebas de software son importantes en el proceso de desarrollo del software ya que en esta etapa es donde se detectan los fallos antes de que sea entregado al cliente. Existen varios tipos de pruebas que propician determinar la calidad del producto, ejemplo de ello son las pruebas de integración; las cuales son las pruebas posteriores a las unitarias y desarrolladas antes de las pruebas de aceptación del cliente. Para la aplicación de dichas pruebas existen varias técnicas y herramientas que posibilitan su automatización como son: la Rational Functional Tester, SOAPU y DB UNIT.

**ABSTRACT**
Software testing is important in the software development process as it is at this stage where faults are detected before it is delivered to the customer. Several types of tests conducive determine product quality, example is the integration tests, these tests are performed after unit testing and tests developed before customer acceptance. For the application of these tests there are several techniques and tools that enable its automation such as: Rational Functional Tester, and DB SOAPU UNIT

# 1. INTRODUCCIÓN

Las industrias del software se ven en constante perfeccionamiento de la calidad del software logrando mejores resultados en la obtención de productos con mayor eficiencia. En Cuba, se ha puesto en práctica la estrategia de prueba de integración descendente debido a las ventajas que esta proporciona, para ello se deben tener una serie de documentos generado por los analistas de las líneas de desarrollo que permita la realización exitosa de la prueba. En el Centro de desarrollo de la Facultad Regional Granma(FRG) se aplica la realización de las pruebas como forma de aseguramiento de la calidad del software. El objetivo de este trabajo es analizar las pruebas de integración, y específicamente determinar cual de las herramientas que permiten su automatización, se considera la más factible.

# 2. CONTENIDO

Pruebas de integración:"es una técnica sistemática para construir la arquitectura del software mientras, al mismo tiempo, se aplican las pruebas para descubrir errores asociados a la interfaz.[1]"Tienen dos objetivos principales como son: descubrir los errores asociados con las interfaces de los módulos y "ensamblar sistemáticamente todos los módulos individuales, para formar varios subsistemas y al final un sistema completo"[2]. La principal dificultad que surge en las pruebas de integración es localizar los errores que se descubren durante el proceso.

## 2.1. Tipos de pruebas y técnicas.

Para la aplicación de las pruebas mencionadas anteriormente se utilizan principalmente técnicas que verifican el correcto manejo de las entradas y salidas del software, es decir, las pruebas funcionales. Además pueden emplearse técnicas que recorren determinadas rutas específicas en la estructura del sistema, como son las pruebas estructurales.

"Existen dos tipos de pruebas de integración:

- Incremental
- No incremental

Dentro de las pruebas de integración incremental existen dos enfoques principales como son:

- Integración descendente (componentes funcionales).
- Integración ascendente (componentes de infraestructura) "[2].

Para la realización de pruebas de integración en la Línea Soluciones de Gestión se propone la utilización del tipo incremental, la misma, permite probar el programa en pequeñas porciones y es más factible para detectar los fallos existentes.

La diferencia entre la pruebas de integración

descendente y ascendente radica, en que esta última define que el programa como entidad no existe hasta que se halla añadido el último módulo, esta prueba ascendente se ve afectada por las características que presente el software y aveces de la planificación del proyecto.

## 2.2. Herramientas

La automatización de las pruebas tiene grandes beneficios, porque probar aplicaciones Web manualmente es una tarea que suele presentar errores. La automatización de las pruebas elimina estos errores; además, como el proceso no requiere la introducción de datos de prueba, es más eficaz y más rápido. A continuación se definen varias herramientas que permiten dicha automatización.

### 2.2.1. IBM Rational Functional Tester

"Es posible configurar los scripts de Rational Functional Tester para que usen datos de orígenes externos, como, por ejemplo, diferentes archivos de Microsoft, Excel. Es utilizada la herramienta para las pruebas a aplicaciones Web, ofrece la posibilidad de cambiar o personalizar los scripts de prueba generados usando las herramientas Java basadas en Eclipse para editar el código de origen Java del script. "[3]

### 2.2.2. SOAPU

Para realizar pruebas de integración de sevicios web se utiliza la herramienta SOAPUI, la misma se puede descargar integrada con Eclipse.

### 2.2.3. DB UNIT

"Permite llevar la base de datos a un estado definido antes de cada prueba y proporciona clases que ayudan a realizar pruebas sobre el contenido de una base de datos. Realiza pruebas de unidad, caja blanca e integración. Es una herramienta de código abierto y libre. Mediante registros de este tipo se pueden generar distintos conjuntos de datos (datasets) en ficheros xml, que pueden servir como semillas para probar el código que accede a la base de datos en situaciones adecuadas para cada prueba. Permite importar y exportar el contenido de la base de datos a ficheros XML". [4] Esta herramienta es una de las más factibles teniendo en cuenta las características especificadas, y es más adaptable según las pruebas realizadas en la Facultad basándose en las características de los proyectos.

### 2.2.4. CRUISE CONTROL

"Es una aplicación de código abierto basado en Java que permite la compilación automática de proyectos Java, utilizando Ant o Maven. Es comúnmente utilizada en integración continua que cada cierto tiempo, o cuando hay cambios en el gestor de versiones (por ejemplo CVS o Subversion), hace una compilación y ejecuta test (más cualquier otra cosa que esté configurada en Ant o Maven) y una vez acaba, presenta el resultado. Esta presentación puede ser en HTML, por correo electrónico, Jabber, entre otros". [4]

# 3. CONCLUSIONES

Aún cuando los módulos de un programa funcionen bien por separado es necesario probarlos conjuntamente: un módulo puede tener un efecto adverso cuando se combinan; por lo cual es imprescindible la aplicación de pruebas de integración que propician que no ocurran errores de este tipo. Luego de expresada las características de algunas herramientas utilizadas para automatizar las pruebas de integración se determinó que la más factible en la Línea Soluciones de Gestión es DB UNIT, por todos los elementos expuestos en el documento.

# 4. REFERENCIAS BIBLIOGRÁFICAS

[1] Pressman, Roger S. Ingeniería del Software. Un enfoque Práctico 6ta Edición. McGraw Hill . Capítulo 13

[2] Rodríguez Tello, Eduardo. Importancia de las pruebas de software. C INVESTAV-Tamaulipas, 30 de marzo 2011.

[3] Automatización de pruebas de integración de aplicación. Citado el 12 de julio 2012. Disponible en [http://www.ibm.com/developerworks/ssa/rational/library/10/automateintegrationtestswithrationalfunctionaltester/index.html]

[4]De la Cruz Leyva, Viana. Set de pruebas para el proyecto Plataforma Libre para la Gestión de Procesos en la Facultad Regional de Granma.Trabajo de Diploma. Universidad de las Ciencias Informáticas Facultad Regional de Granma, 2012.

## 5. BIBLIOGRAFÍA

Estrategia de prueba de integración descendente a los subsistemas del sistema integral de Gestión CEDRUX en el proyecto ERP-Cuba. Citado el 12 de julio del 2012. Disponible en. [http://www.informaticahabana.cu/node/750]

IBM Rational.Disponible en.[http://www.insight.com.uy/ibm.html]

Pruebas de software. Citado el 2 de noviembre 2001. Disponible en:
[http://www. angelfire.com/my/ jimena/ingsoft /guia 10.html]