

Tipo de artículo:
Temática: Ingeniería y gestión de software
Recibido: dd/mm/aa | Aceptado: dd/mm/aa | Publicado: dd/mm/aa

Experiencias en la aplicación de pruebas de software a la Plataforma GENESIG y sus aplicativos

Design of a strategy of tests for the platform GENESIG and his applicative ones

Zaylí Rodríguez Luis^{1*}, Liliam Rodríguez Landestoy

¹ GEySED. Departamento de Señales Digitales. Universidad de las Ciencias Informáticas, Carretera a San Antonio de los Baños, km 2½, Torrens, Boyeros, La Habana, Cuba. CP.: 19370

*Autor para la correspondencia: zluis@uci.cu

Resumen

Un Sistema de Información Geográfica es una tecnología de manejo de información georeferenciada, que provee las funciones y las herramientas necesarias para almacenar, analizar y desplegar la información espacial. Su uso se hizo más frecuente gracias al desarrollo de una tecnología informática adecuada, que fomentó la aparición de productos SIG en el mercado. La Universidad de las Ciencias Informáticas dispone de una plataforma para el ensamblaje de los SIG. Por su impacto económico y social se le atribuye gran importancia a este producto, que carece de un proceso que sea capaz de validar la calidad tanto de la plataforma como de todos sus aplicativos, el cumplimiento de los requisitos y especificaciones para ser entregado al cliente, por lo cual se convierte en una necesidad para conocer los posibles errores sobre los que se deben trabajar para asegurar la calidad de estos sistemas. Para suplir esta carencia fue diseñada una estrategia de pruebas, que permitió organizar y planificar las actividades llevadas a cabo durante todo el proceso. Para la siguiente investigación se utilizaron métodos teóricos y empíricos, los cuales permitieron profundizar en el estudio del flujo de trabajo de pruebas como parte del desarrollo del software. Se tomó como muestra el Sistema de Información Geográfica de la Universidad, al que se le aplicaron un conjunto de pruebas y revisiones, las cuales permitieron detectar gran cantidad de errores que deben ser corregidos para certificar la calidad de dicho producto.

Palabras clave: calidad, estrategia, pruebas, sistemas de información geográfica.

Abstract

A Geographic Information System is a technology of georeferenced information, which provides functions and tools to store, analyze and display spatial information. Its use became more prevalent with the development of appropriate computer technology, which encouraged the development of Geographic Information System products on the market. At the University of Information Sciences built a platform for the assembly of the Geographic Information System. For its economic and social impact attaches great importance to this product because there is a process that is capable of validating the quality of both the platform and all applications not meeting the requirements and specifications to be delivered to customer becomes a necessity to know the possible errors on what should work to ensure the quality of these systems. Responding to this problem, we designed a test strategy that allowed the organization and planning activities conducted throughout the process. For the following investigation were used within the scientific methods and empirical theoretical methods, which allowed further study of related workflow tests as part of software development. Sample was taken as the geographical information system of the University. At the

same were applied to a set of tests and reviews, which allowed us to detect many errors to be corrected to certify the quality of the product.

Keywords: *geographic information systems, quality, strategy, testing.*

Introducción

El inicio del desarrollo de los Sistemas de Información Geográfica (SIG) comienza en los años 60 relacionándose con diversos campos de investigación. Surge bajo condiciones ambientales, es decir, era necesario estudiar las características físicas y químicas de la atmósfera para contribuir con su protección. A principios de los 80, se hizo más frecuente gracias al desarrollo de una tecnología informática adecuada, que fomentó la aparición de estos productos en el mercado. El surgimiento de los SIG puso en manos de los investigadores una herramienta que permitía el procesamiento espacial de gran cantidad de información en forma de capas. Cuba no ha quedado exenta de este desarrollo, en 1987 surge el Instituto de Geografía de la Academia de Ciencias de Cuba "El SIG de Cuba" con el objetivo fundamental de actualizar el atlas nacional de Cuba. En las últimas décadas su uso se ha incrementado enormemente pasando del total desconocimiento a la práctica cotidiana. Todo esto se debe a determinados factores que han fomentado su desarrollo. Entre los que se encuentran: La reducción de costos de los equipos informáticos, el uso de la información geográfica desde múltiples ámbitos para ayudar la toma de decisiones y la sociedad de la información que demanda cada vez más cualquier tipo de datos. Por todo esto se considera que están de total actualidad y son muchas las áreas en las que intervienen.

Un SIG es un sistema dinámico computacional que utiliza datos geográficos, los cuales permiten mapear cualquier tipo de información almacenada en planillas o base de datos, para lograr un mejor análisis. El objetivo principal de un SIG es administrar, modelar y analizar información relacionada con datos espaciales. Actualmente muchos de los problemas sociales que padece el mundo contienen una dimensión geográfica valorada de crítica como resultado de la pobreza y se manifiesta en la contaminación, superpoblación, los desastres naturales y otros, por ello; para la búsqueda de soluciones técnicas es necesario el acceso a la información y el trabajo de especialistas en equipos multidisciplinarios. La tecnología SIG, además de una disciplina académica; es una industria importante ya que sus aplicaciones son actualmente enseñadas y expandidas en todas las universidades del mundo.

Los jóvenes profesionales cubanos son cada vez más conscientes de las ventajas que proporciona que un sistema almacene y manipule información geográfica, ya que cerca del 80% de la información tratada en empresas e instituciones está relacionada con localizaciones geográficas o coordenadas espaciales (Cueva, 2009). En estos sistemas es elevado el nivel de responsabilidad e importancia, por lo que se hace necesaria la presencia de la calidad de software, que no es más que la forma de asegurar la concordancia entre los requisitos funcionales y los de rendimiento con los estándares de desarrollo (Cueva, 2009). Las actividades planificadas y sistemáticas que contribuyen a que el producto logre satisfacer los requisitos expuestos aseguran la calidad del software.

Para garantizar la calidad adecuada de un SIG es necesario que se les realicen pruebas de software; identificando los posibles errores de implementación, calidad o usabilidad. Y la detección oportuna de fallos en el software con el fin de minimizar los costos de reparación. Si se detecta una falla con rapidez, su corrección será mucho más barata.

La Universidad de las Ciencias Informáticas (UCI) desde su surgimiento, ha obtenido significativos logros informáticos y trabaja en la formación integral de profesionales que respondan a las necesidades del progreso

científico técnico y socioeconómico de la sociedad; para ello cuenta con varios Centros Productivos, que a su vez están conformados por diferentes Departamentos que contienen los Proyectos especializados en la elaboración de software. En el centro GEySED de la facultad 6, ideado para el desarrollo de software relacionados con la Geoinformática y la captura de señales digitales, se encuentra el Departamento Geoinformática donde se desarrolla el proyecto GENESIG, encargado de la elaboración de SIG mediante una plataforma que lleva ese mismo nombre.

El mismo es sustentado por un personal altamente calificado con la función de aplicar esta ciencia. Esta herramienta para el ensamblaje de SIG es interoperable con soluciones existentes, integradas con software de gestión para la toma de decisiones. En los últimos tiempos se ha incrementado exponencialmente el uso de aplicaciones de este tipo para la toma de decisiones, lo cual está íntimamente relacionado con el incremento de la efectividad y la disminución de los costos que se obtienen con la puesta en marcha de un sistema con estas características. Constituye un producto de gran actualidad y con grandes expectativas a nivel mundial. Por el impacto económico notable en relación con los SIG actuales se le atribuye un valor incondicional a este sistema, el cual no cuenta con un proceso que sea capaz de validar la calidad tanto de la plataforma como de todos sus aplicativos, además del cumplimiento de los requisitos y especificaciones para ser entregado al cliente.

Esta investigación se traza como objetivo general: Desarrollar una estrategia de pruebas para la plataforma GENESIG y sus aplicativos mediante un plan de pruebas que permita certificar la calidad de estos productos. Con la finalidad de orientar, servir de guía al proceso de investigación y la solución del problema se parte de la siguiente hipótesis: si se lleva a cabo un correcto diseño y aplicación de pruebas de calidad a la plataforma GENESIG y sus aplicativos se podrá verificar y revelar la calidad de estos productos.

1. Factores que determinan la calidad del software

Es de gran importancia profundizar en el conocimiento de los factores que miden y determinan la calidad del software, ya que constituyen puntos claves en el desarrollo de los mismos.

Se clasifican en tres grupos:

- Operaciones del producto: características operativas.
 - Corrección
 - Fiabilidad
 - Eficiencia
 - Integridad
 - Facilidad de uso
- Revisión del producto: capacidad para soportar cambios
 - Facilidad de mantenimiento
 - Flexibilidad
 - Facilidad de prueba
- Transición del producto: adaptabilidad a nuevos entornos
 - Portabilidad
 - Reusabilidad
 - Interoperabilidad (Pressman, 2005)

2.1 ¿Qué son las pruebas de software?

Las pruebas de software conforman el conjunto de actividades que originan los procesos capaces de detectar los errores en un producto. Cualquier aplicación que haya sido construida necesita de las pruebas de software y se deben tener en cuenta los niveles de las mismas para su correcta ejecución. (Rojas, 2007). Este proceso permite indagar en cada una de las fases del software, para tales fines realiza una búsqueda consistente y eficaz, que posibilita probar cada una de las funcionalidades del sistema. Es importante aclarar que la realización de las pruebas asegura la presencia de errores en caso que éstos existan, pero nunca podrá ser capaz de garantizar que el producto esté libre de defectos (Booch, 2000).

2.2 Características a probar

Cuando se va a probar un software es necesario tener en cuenta varias características dependiendo de las condiciones del producto y además del objetivo que persiguen las pruebas a realizar. Luego de analizar las propiedades del aplicativo SIGUCI se determinó que teniendo en cuenta que el cliente pretende asegurarse que las funcionalidades en el sistema se realicen de manera correcta, se convierte en una función principal verificar que:

- Sea correcta la implementación de los requisitos funcionales.
- Cuenten con una interfaz amigable, comprensible y factible para el usuario.
- Presente una documentación legible y entendible ante los usuarios.
- Tenga implementado correctamente mecanismos que garanticen la seguridad de la aplicación.
- El rendimiento y la carga de trabajo que puede soportar el sistema.

Pudiendo observar que el software a probar es un sistema de información geográfica es necesario verificar que además de las condiciones anteriores sea capaz de gestionar correctamente toda la información que almacena. Para esto es imprescindible que el sistema cuente con:

- Correcta entrada de la información.
- Eficiente almacenamiento y manipulación de la información.
- Calidad en la representación y salida de los datos georeferenciados.
- Válida utilización de la fuente cartográfica.

Las pruebas fueron aplicadas en el nivel de sistema, lo que permite probar el funcionamiento del producto SIGUCI en su totalidad, después de integrados sus módulos y a través de la interfaz de usuario. Se define este nivel debido a que se pretende probar la integridad y funcionalidad del sistema, tratando de encontrar la mayor cantidad de errores. Las técnicas de pruebas de dicho nivel permiten que el sistema se ejercite, además se comprueba la integración de la información y el funcionamiento de las interfaces.

Cada nivel de prueba contiene métodos y técnicas las cuales están diseñadas con el objetivo de encontrar posibles errores del software en su respectivo nivel. Dentro de los tipos de prueba que se realizan en el nivel de Sistema, las escogidas que se aplicaron al proyecto SIGUCI fueron las siguientes:

- Pruebas exploratorias
- Pruebas Funcionales
- Pruebas de seguridad
- Pruebas de carga y estrés

– Pruebas de usabilidad

Debido a que la plataforma GENESIG y sus aplicativos proveen las funciones y las herramientas necesarias para almacenar, analizar y desplegar la información geográfica, fue necesario realizar pruebas que permitan garantizar la correcta ejecución de las principales funcionalidades de los SIG. Por lo explicado anteriormente se redactó una lista de chequeo, la cual evalúa los siguientes parámetros que se muestran a continuación en el siguiente diagrama:

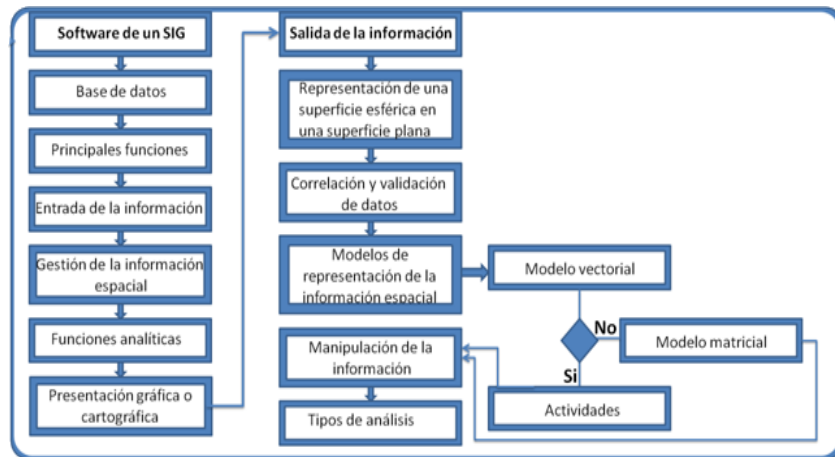



Figura 1. Diagrama de las Pruebas básicas de las principales funciones de un SIG.

1.3 Ejecución de las pruebas

Esta fase es de gran importancia y constituye la esencia de la estrategia de pruebas. Aquí es donde se aplican las pruebas que fueron escogidas anteriormente al aplicativo SIGUCI tomado como muestra. Cada una de las actividades se describirá a continuación.

Tabla 1. Actividades de la sexta fase de la estrategia de pruebas.

Fase	Actividades
	1- Realizar pruebas exploratorias
	2- Realizar pruebas funcionales
	3- Realizar pruebas de usabilidad
	4- Realizar pruebas de carga y estrés
	5- Realizar pruebas de seguridad

1.3.1 Pruebas exploratorias

Para la realización de las pruebas exploratorias se tuvieron en cuenta algunos aspectos de la usabilidad con el objetivo de hacer más factible el resultado de estas pruebas. El probador que llevó a cabo esta actividad tuvo una preparación por parte del líder del proyecto sobre el sistema. Además participó en talleres sobre los SIG, pruebas, revisiones y aplicaciones web. Estas pruebas fueron desarrolladas mediante una interacción exploratoria del probador con el sistema.

1.3.2 Pruebas funcionales

Las pruebas funcionales que se le realizaron a SIGUCI fueron pruebas de caja negra. Un probador y el jefe de proyecto fueron los protagonistas de la actividad. Estas pruebas fueron basadas en los diseños de casos de prueba que se explicaron con anterioridad, donde se recogen los escenarios correspondientes a los casos de uso del sistema y las clases de datos utilizadas para probar estos escenarios.

Para la realización de estas pruebas se ejecutó cada caso de uso, flujo de caso de uso o función usando datos válidos e inválidos. Se eligieron correctamente los valores en las entradas abarcando el máximo número de combinaciones de datos de entrada sin que el número de casos de pruebas sea muy elevado. Se realizaron 2 iteraciones, cada una tuvo una duración de 7 días. Mediante estas pruebas se verificaron todos los requisitos especificados por el cliente.

1.3.3 Pruebas de usabilidad

Las pruebas de usabilidad fueron ejecutadas con usuarios reales, a los cuales se les presentaron un conjunto de tareas a realizar con el prototipo del sistema. Estas tareas fueron basadas en una lista de chequeo con interrogantes relacionadas con los factores de usabilidad en aplicaciones web. Los comentarios de los usuarios que participaron fueron recogidos para optimizar el resultado de dicha prueba.

Como complemento a esta prueba se realizó la evaluación de un experto en usabilidad. El mismo proporcionó una información completamente diferente a la que se obtuvo de los usuarios finales, pues es más viable y precisa.

1.3.4 Pruebas de carga

Las pruebas de carga fueron realizadas usando la herramienta Jmeter. Es una herramienta automatizada para aplicaciones web muy versátil, capaz de simular usuarios, ahorrar recursos y tiempo, es gratis y libre. Esto condujo a tener en cuenta determinados requisitos para que los resultados sean satisfactorios.

Sobre el aplicativo SIGUCI que fue la aplicación sometida a las pruebas fue necesario conocer en cuanto al software:

- Datos de software de la aplicación.
- Tipo de conexión y velocidad.

Sobre hardware:

- Datos sobre la velocidad y la capacidad del servidor de la aplicación.
- Distancia del servidor al laboratorio de pruebas.

Este proceso consta de dos elementos fundamentales:

- Grabación de los escenarios de pruebas.
- Confección de los planes de pruebas.

Para la realización de la grabación de los escenarios de prueba es necesario configurar el navegador Firefox con el proxy definido en el JMeter, y luego se pone a correr la aplicación realizando las tareas en ella que se desea que sean grabadas por el programa para su posterior simulación. De esta forma queda grabado el escenario de prueba. Para la confección de los planes de prueba se utilizan los elementos definidos en el procedimiento.

Para interpretar los resultados del JMeter se debe conocer:

- Cantidad de usuarios concurrentes que necesita soportar la aplicación.
- Tiempo de respuesta en situaciones de carga de una transacción.

Además el proyecto en el documento de los casos de prueba debe describir los escenarios significativos que serán sometidos a la prueba.

1.3.5 Pruebas de seguridad

Las pruebas de seguridad al producto SIGUCI fueron basadas en la guía de OWASP (Proyecto Abierto de Seguridad para Aplicaciones Web). Se utilizó esta guía ya que permite mejorar las habilidades para probar. Además varias de las pruebas aquí descritas no son complicadas ni requieren habilidades especiales o herramientas. Con OWASP se pueden expandir los casos de comprobación aplicados a los programas. También se detectan vulnerabilidades ahorrando tiempo y recursos considerables.

A la aplicación SIGUCI se le realizaron pruebas de intrusión, las mismas son las encargadas de comprobar la seguridad de una red. Esta técnica es el arte de comprobar un sistema en ejecución, sin conocer su funcionamiento interno, con el objetivo de encontrar vulnerabilidades de seguridad. Para el desarrollo de estas pruebas se realizó un análisis activo de la aplicación en busca de cualquier debilidad, fallos técnicos o vulnerabilidades. Las incidencias de seguridad encontradas fueron recogidas en un documento junto con una propuesta para su mitigación o una solución técnica.

Las pruebas que se le aplicaron a SIGUCI fueron divididas en 4 subcategorías.

1.3.6 Pruebas de autenticación

Transmisión de credenciales a través de un canal cifrado (OWASP-AT-001)

Se comprobó que los datos de autenticación del usuario sean transferidos a través de un canal cifrado para evitar que sean interceptados por usuarios maliciosos. Con esto se pretende entender si los datos viajan no cifrados desde el servidor web al servidor o si la aplicación utiliza los mecanismos de seguridad apropiados.

Pruebas de diccionario sobre cuentas de Usuario o cuentas predeterminadas (OWASP-AT-003)

Se realizaron varias combinaciones de nombre de usuario/contraseña predeterminadas y que pueden ser empleadas para obtener acceso a la infraestructura de la red interna para obtener privilegios y robar datos. Esta comprobación se realizó con una herramienta automatizada llamada Brutus. Se elaboraron dos documentos de textos con los posibles usuarios y contraseñas, los cuales fueron cargados por esta herramienta. De esta forma se comprobó cualquier posible entrada al sistema con estos usuarios y contraseñas.

1.3.7 Gestión de sesiones:

Primeramente se comprobó la presencia de la función de cierre de la sesión, comprobando que la aplicación proporciona un botón de cierre de sesión y que esté presente en las páginas que requieran autenticación. Mediante esta prueba también se comprueba lo que le ocurre a los testigos de sesión cuando la función de cierre es invocada. Se verificó que las cookies fueran borradas. También se realizó la prueba de cierre de sesión por tiempo expirado. Se comprobó si existe un tiempo de expiración registrando un usuario en la aplicación y esperar un tiempo determinado para ver si se dispara el cierre de sesión por tiempo. Luego que haya pasado este tiempo todos los testigos de sesión deberán ser destruidos.

1.3.8 Pruebas de elementos básicos de un SIG

Estas pruebas fueron ejecutadas directamente a la aplicación. Estuvieron presentes un implementador del sistema, el analista principal, un diseñador de base de datos y el jefe de proyecto junto al probador encargado de la actividad. Se recibió con anterioridad una capacitación sobre las características fundamentales, las funciones, la composición y otras informaciones generales de los SIG. Mediante la lista de chequeo elaborada se le dio respuesta a cada una de las interrogantes planteadas. Se identificaron las deficiencias e inconformidades en un documento, que posteriormente fue revisado y validado por expertos en el tema.

3. Resultados obtenidos de la ejecución de las pruebas

Resultados obtenidos de la ejecución de las pruebas

Resultado de las pruebas aplicadas a SIGUCI

Pruebas exploratorias

Con la aplicación de estas pruebas se encontraron 7 no conformidades que no habían sido detectadas, lo que ratifica su importancia. La lista de recomendaciones fue entregada al equipo de desarrollo y se estableció como plazo para erradicar las no conformidades un tiempo estimado de 72 horas.

Los principales problemas encontrados fueron:

- La aplicación no se muestra en el navegador: Internet explorer.
- Existen espacios en blanco en algunas interfaces.
- No se identifica al usuario cuando se encuentra en la aplicación.

Pruebas de caja negra

Las pruebas funcionales al sistema se realizaron en dos iteraciones como se especificó en capítulos anteriores, para verificar que las funcionalidades implementadas fueron las acordadas con el cliente y que respondían correctamente a sus necesidades. La siguiente tabla muestra el resumen de los defectos encontrados como resultado de estas pruebas.

Tabla 2. Resumen estadístico de las no conformidades en las pruebas funcionales.

Aplicación	Iteración	No conformidades	Significativas	No significativas	Recomendaciones
	1	7	6	1	4
	2	5	4	1	4
Total	2	12	10	2	8

De las no conformidades encontradas, las que están clasificadas como significativas están relacionadas con problemas de funcionalidad del sistema, que en ocasiones no coincidían con las descripciones de los casos de uso a los que responden. Los errores clasificados como no significativos están relacionados con el idioma, principalmente textos o mensajes de errores. Las recomendaciones que fueron redactadas perseguían como objetivo principal proponer mejoras para el sistema y prevenir la ocurrencia de fallas en un futuro. Una vez que el probador le hizo entrega al equipo de desarrollo los defectos encontrados en una iteración contaban con 72 horas para resolverlos.

Los principales problemas que se detectaron durante la aplicación de estas pruebas son:

- No se muestran correctamente todos los datos del edificio que busca el usuario.
- No se encuentran las imágenes en las descripciones de algunas entidades.
- La interfaz correspondiente a la sección “Mapa temático” no se muestra en la aplicación, por lo que no se puede probar ningún escenario de esta sección.
- Algunos errores que se muestran para especificar los valores que el usuario debe entrar son descritos en inglés.

Pruebas de usabilidad

Las pruebas de usabilidad permitieron detectar fallas e ineficiencias en la aplicación. En la tabla que se muestra a continuación se encuentra un resumen de las no conformidades encontradas durante la realización de estas pruebas.

Tabla 3. Resumen estadístico de las no conformidades en las pruebas de usabilidad.

Aplicación	Iteraciones	No conformidades	Impacto Alto	Impacto Medio	Impacto Baja	Recomendaciones
	1	24	20	4	0	24
	2					

Los principales problemas que se encontraron se especifican a continuación.

- No cuenta con una ayuda para los usuarios que interactúan con el sistema.
- El sitio no presenta contactos para conocer a los administradores del mismo.
- Existen mensajes de error en inglés.
- La aplicación no contiene la fecha.
- No se brinda la opción de aumentar o disminuir el tamaño de letra.
- El sitio no presenta enlaces.
- Contiene demasiados elementos en el menú de navegación.

Pruebas de carga y estrés

Durante las pruebas de carga, analizando los valores de la cantidad de peticiones y rendimiento se puede concluir que el sistema responde satisfactoriamente. La prueba se realizó con 50 usuarios concurrentes y mantuvo un rendimiento estable. Los resultados obtenidos demuestran que las funcionalidades cumplen con los requerimientos establecidos exceptuando la funcionalidad: Buscar persona, la cual alcanzó un 100% de error con un rendimiento desfavorable.

La prueba de estrés fue realizada con el objetivo de conectar 150 usuarios concurrentes, para esto fue llevada a cabo con 15 usuarios que se conectaban cada 5 segundos en un ciclo que se repite 10 veces, intentando lograr 150 usuarios en el último ciclo. Al culminar la prueba se pudo concluir que el rendimiento del sistema no fue estable, alcanzando valores inaceptables y posibilitando un lento avance de la aplicación que tardó 2 minutos en reponerse. Por todo lo anterior el sistema soporta hasta 100 usuarios concurrentes presentando un buen rendimiento y un tiempo de respuesta dentro de los parámetros normales. Un valor por encima de este no se acepta.

De forma general los problemas principales encontrados durante la aplicación de estas pruebas son:

- Errores en vínculos de la aplicación.
- Errores en el sistema por la carga en la herramienta.
- Problemas de resistencia de la Base de Datos.
- Problemas del navegador seleccionado.

Las pruebas realizadas con el navegador Firefox presentan un por ciento de errores superior a cuando son realizadas con el internet explorer, ya que ocurren errores en elementos del mismo que no pueden ser reconocidos cuando se le da respuesta a las peticiones.

A continuación se muestran los resultados obtenidos en las pruebas de carga y estrés.

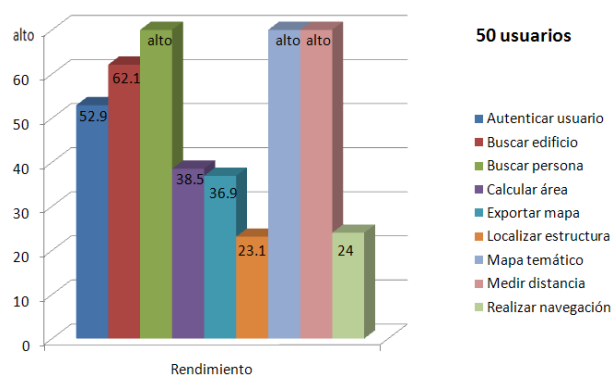


Figura 3. Valores de rendimiento en las pruebas de carga.

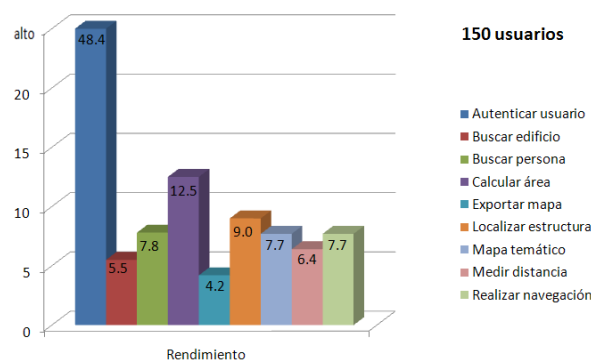


Figura 4. Valores de rendimiento en las pruebas de estrés.

Pruebas de seguridad

Mediante las pruebas de Transmisión de credenciales a través de un canal cifrado se pudo detectar que el aplicativo SIGUCI no utiliza un protocolo seguro para transmitir los datos cifrados, lo cual da paso a que no estén protegidos por algún atacante. Las pruebas de diccionario sobre cuentas de Usuario o cuentas predeterminadas permitieron detectar que la aplicación no es sensible a entradas de usuarios.

De los usuarios y contraseñas que se encontraban en los archivos cargados por la herramienta Brutus ninguno de ellos fueron capaces de entrar al sistema. Esto demuestra que no existen deficiencias en el control de la entrada de usuarios al sistema. Al realizarse las pruebas de gestión del Caché de Navegación y de salida de sesión se pudo detectar una falla grave en la aplicación. No cuenta con un botón de cierre de sesión por lo que las cookies no son eliminadas y

muestran datos importantes del usuario. Tampoco presenta un tiempo de expiración de sesión, esto contribuye aún más a la presencia de inseguridad en este sistema.

Pruebas de elementos básicos de un SIG

Al aplicar la lista de chequeo expuesta en el capítulo anterior, la cual permite verificar la correcta implementación de las principales funciones que deben estar presentes en un SIG, se pudo detectar una gran deficiencia de estos elementos en el SIGUCI. Se realizó una sola iteración de esta prueba, donde se recogieron 22 no conformidades. A continuación se muestra una gráfica que expresa los resultados alcanzados.

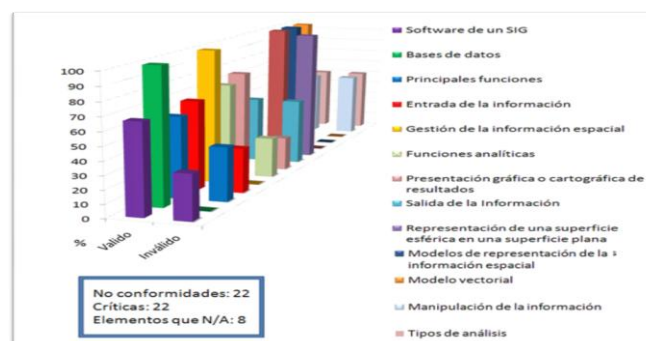


Figura. 5. Pruebas de elementos básicos de un SIG.

Como se puede apreciar en la figura anterior, los elementos: Base de Datos, Gestión de la información espacial, Correlación y validación de datos, Modelos de representación de la información espacial y Modelo vectorial fueron evaluados con el 100% de validez. Esto significa que el sistema cuenta con una correcta implementación de estas funcionalidades, ya que no se encontraron no conformidades al respecto.

Además se puede observar que en cuanto a la Salida de la información, la Manipulación de la información y los Tipos de análisis se obtuvo un 50% de errores. Con este resultado se demuestra que solo la mitad de estos elementos se encuentran funcionalmente correctos.

Los principales problemas encontrados durante la realización de estas pruebas son:

- **Se utiliza una fuente cartográfica desconocida.**

Esto constituye un error grave en cualquier SIG, ya que al no conocer la fecha de la fuente cartográfica es imposible afirmar la actualidad de la misma. De esta forma existe la posibilidad que todas las funcionalidades y bondades que brinda el sistema sean incorrectas, ya que este elemento es fundamental en el desarrollo de un SIG.

- **No permite encontrar un lugar donde se cumplan ciertas condiciones.**

Esta función forma parte de los requisitos principales que todo SIG debe cumplir. Es importante que el usuario pueda identificar un lugar específico dadas ciertas condiciones como parámetro.

- **No se utiliza ninguna técnica para la teledetección y fotointerpretación de imágenes.**

Estas técnicas permiten obtener información en formato digital directamente, además de incluir un grupo de funciones que posibilitan el intercambio de ficheros con otros programas y entornos diferentes, por lo que se hace necesaria su presencia en los SIG.

- **No presenta funciones analíticas como: vecindad, conectividad y proximidad.**

La iteración lógica de funciones analíticas permite la creación de modelos para simular distintos procesos y evaluar diferentes aspectos que nos ayuden en la toma de decisiones y en la selección de alternativas.

Conclusiones

Se realizó una descripción explícita y elocuente de los artefactos, actividades y todos los aspectos relacionados con el flujo de trabajo de pruebas como parte del desarrollo de software. Se diseñó una estrategia de pruebas para la plataforma GENESIG y sus aplicativos, la cual logró una mayor organización en las actividades llevadas a cabo durante el proceso de la ejecución de las pruebas. Se verificó la correcta implementación de los requisitos planteados por el cliente encontrando errores significativos que no fueron detectados por los desarrolladores y que además empañan la buena calidad y estructura tanto del documento como de la aplicación.

Referencias

- CUEVA JUAN MANUEL LOVELLE. Calidad del Software [Publicación periódica].- España: [s.n.], 2009.- s.n.
- PRESSMAN ROGER, S. Ingeniería de Software: Un enfoque Práctico [Libro].- 2005.
- BOOCH JACOBSON, IVAR y RUMBAUGH GRADY, JAMES. El Proceso Unificado de Desarrollo de Software. Madrid: s.n., 2000. 84-7829-0362.
- JOHANNA ROJAS ROJAS Y EMILIO JOSÉ BARRIOS. Investigación sobre estado del arte en diseño y aplicación de pruebas de software. [en línea] 2007 [Consultado el: 25 de noviembre de 2010]. Disponible en: [\[http://www.udistrital.edu.co\]](http://www.udistrital.edu.co).