

Herramienta computacional para la documentación de pruebas de software enmarcado en actividades de investigación

Computer tool for software testing documentation under research activities

Nelson Enrique Leon Martinez^{1*}, Juan Luís Aguilar Garcia², Edgar Fernando Vega Morales³, Luis Carlos Gomez Florez⁴

Ingeniería de Sistemas, Universidad Industrial de Santander, Norte de Santander, Colombia

leonmartinez82@yahoo.com
 juanaguilargarcia20@gmail.com
 ferchogmv15@gmail.com
 lcgomezf@uis.edu.co

Resumen— El presente artículo describe el diseño de una herramienta computacional para la documentación de pruebas de software. La herramienta permite gestionar documentos como el plan maestro de pruebas; los planes de nivel, casos y procedimientos de prueba; reporte de anomalías y reporte de pruebas, que hacen parte de una adecuada documentación de pruebas de software. Utilizando diagrama de paquetes y casos de uso se muestra la funcionalidad disponible en la herramienta, la cual ha sido diseñada a partir de la norma IEEE 829.

Palabras clave— *Calidad, Documentación, IEEE 829, Pruebas, Software.*

Abstract— This paper describes the design of a computational tool for software test documentation. The tool manages documents as the master plan test; level plans, cases and procedures test; anomalies report and test reports, which are part of a proper documentation of software testing. Using package and use cases diagrams the functionality available in the tool is described, which has been designed from the IEEE 829 Standard.

Key Word — *Documentation, IEEE 829, Quality, Software testing.*

I. INTRODUCCIÓN

Teniendo en cuenta que las pruebas de software constituyen uno de los procesos más importantes en el desarrollo de software ya que permiten validar si una aplicación cumple con los requisitos planteados por el usuario, verifican que fue construido correctamente, detectan los puntos críticos para que sean corregidos, pueden proporcionar información importante sobre la calidad de un producto y a su vez evitan a futuro pérdidas económicas, de personal, de información, etc., se puede entender el gran esfuerzo que hacen hoy día las

organizaciones por llevar a cabo este tipo de prácticas destinando para ello entre el 30 y 40 por ciento del esfuerzo de un proyecto [1].

Incluida en la realización de pruebas existe una actividad muy importante que es la documentación de estas. Prácticamente todos los desarrolladores podrían afirmar que efectúan pruebas durante el desarrollo de sus aplicaciones ya sea haciendo uso de las técnicas y métodos existentes o simplemente al ejecutar la aplicación y verificar su funcionamiento; pero estas pruebas no pueden quedar a la deriva por lo cual es necesario poseer una evidencia de las actividades de pruebas realizadas y los resultados obtenidos. Por lo tanto se puede decir que la documentación es inherente a las pruebas, porque sin ella no se puede constatar lo que se ha hecho y no se puede valorar la calidad del software probado.

Por otro lado, en [2], [3] y [4] se describe el diseño de una herramienta computacional que sirve como soporte para la evaluación de la calidad de software desde tres puntos de vista: Proyecto, Producto y Proceso. Esta herramienta está orientada para que sea utilizada en la valoración de la calidad de las aplicaciones realizadas en los grupos de investigación que desarrollan software como parte de sus actividades y donde los recursos de personal, tiempo y dinero son escasos para implementar completamente estándares internacionales para evaluar su calidad. No obstante, si se desea evaluar de forma más completa la calidad de un producto software, no se puede dejar a un lado las pruebas de software y su respectiva documentación como evidencia visible de las actividades de prueba realizadas.

Por lo anterior, se hace necesario el desarrollo de una nueva versión del software en el que se incluya un módulo para la documentación de pruebas, a través del cual se pueda tener un soporte de las pruebas realizadas y se pueda sustentar los resultados de las evaluaciones realizadas. A continuación se presenta una descripción de la estructura fundamental del estándar tomado como base para diseñar la herramienta. Luego se muestra la estructura funcional de la misma utilizando diagramas de paquetes y de casos de uso y por último se presentan las conclusiones y las recomendaciones.

II. DOCUMENTACIÓN DE PRUEBAS SOFTWARE.

A. Pruebas de software

Las pruebas de software son procesos realizados concurrentemente a través de las etapas del desarrollo de software que utiliza y mantiene el testware y cuyo objetivo es apoyar la disminución del riesgo de aparición de fallas y faltas en operación [5].

Dentro del marco de las pruebas de software se encuentran las denominadas estrategias y técnicas, que son prácticas muy importantes para lograr mejores resultados durante el proceso de pruebas. Las estrategias de las pruebas son importantes, ya que si un software se prueba sin tenerlas en cuenta, seguramente se desperdiciaría tiempo, se dedicaría un esfuerzo innecesario y lo que es peor, puede que no se detecte ningún error [1] (ver figura 1).

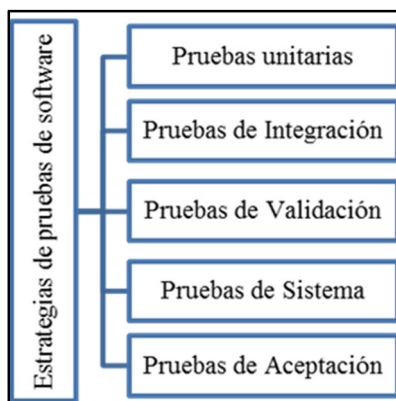


Figura 1. Estrategias de pruebas de software

Las técnicas de las pruebas de software no son más que un conjunto de procedimientos a seguir para obtener los mejores resultados al momento de realizar las pruebas de software. Dentro de estas técnicas se destacan las de caja blanca, negra, gris y las pruebas de entornos especializados [1](ver figura 2).

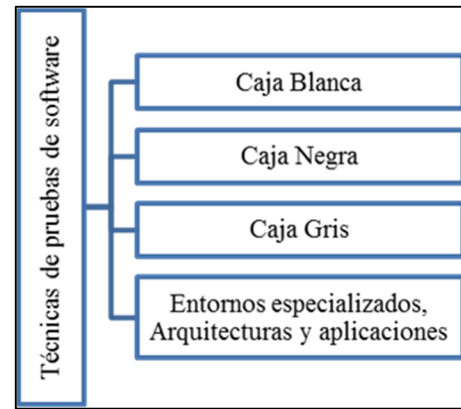


Figura 2. Técnicas de pruebas de software.

B. Documentación de pruebas con la norma IEEE 829.

Al momento de querer efectuar las actividades de documentación de pruebas son útiles como guía varias normas y estándares internacionales que están dispuestos para ello, dentro de los cuales se encuentran los siguientes: Estándar británico BS 7925[6], IEEE 829 Documentación de pruebas [7], ISO / IEC 29119 software testing[8], IEEE 1008 Pruebas de Unidad[9], entre otros. No obstante la norma más efectiva para una adecuada documentación es la IEEE 829 ya que ha sido diseñada especialmente para la documentación de pruebas y por medio de su utilización se puede abarcar de una forma más amplia y precisa las actividades de documentación de pruebas, aprovechando al máximo los datos que podemos obtener de las pruebas que se realizan.

Esta norma establece diez documentos básicos que deben ser generados durante el proceso de pruebas, que son: Plan maestro de pruebas, Plan(s) de pruebas de nivel, Diseño de pruebas, Casos de prueba, Procedimiento de la prueba, Registro de prueba, Reporte de anomalías, Reporte provisional del estado de la prueba, Reporte de pruebas, Reporte maestro de pruebas, como se resume en la figura 3.

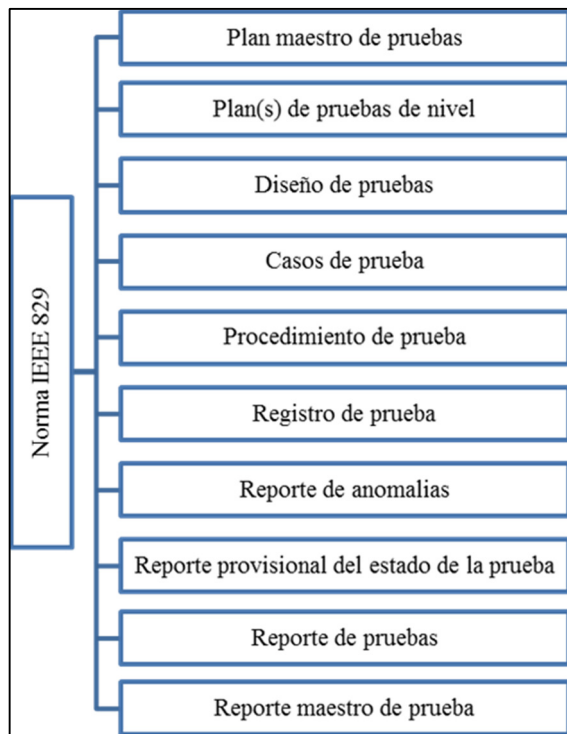


Figura 3. Documentación de pruebas software, Norma IEEE 829.

Para determinar cuáles de estos documentos se deben generar o no como producto de las actividades de prueba que se realicen, la norma sugiere dos criterios de selección. El primer criterio consiste en establecer un nivel de integridad para el software y, basados en el nivel seleccionado, la norma sugiere los documentos mínimos que se deben generar con la realización de las pruebas. A continuación se muestra una tabla en la cual se observan los diferentes niveles de integridad y los respectivos documentos que se deben generar como mínimo en cada uno de ellos.

Nivel de integridad	Documentación de pruebas
1 Insignificante	Plan(es) de pruebas, Diseño de pruebas, Casos de prueba, Procedimiento de la prueba, Registro de prueba, Reporte de anomalías, Reporte de pruebas.
2 Marginal	Plan(es) de pruebas de nivel, Diseño de pruebas, Casos de prueba, Procedimiento de la prueba, Registro de prueba, Reporte de anomalías, Reporte provisional del estado de la prueba, Reporte de pruebas.
3 Crítico	Plan maestro de pruebas, Plan(es) de pruebas, Diseño de pruebas, Casos de prueba, Procedimiento de la prueba, Registro de prueba, Reporte de anomalías, Reporte provisional

	del estado de la prueba, Reporte de pruebas, Reporte maestro de pruebas.
4 Catastrófico	Plan maestro de pruebas, Plan(es) de pruebas, Diseño de pruebas, Casos de prueba, Procedimiento de la prueba, Registro de prueba, Reporte de anomalías, Reporte provisional del estado de la prueba, Reporte de pruebas, Reporte maestro de pruebas.

Tabla 1. Niveles de integridad de la documentación de pruebas

El segundo criterio es el de libre decisión, es decir, los documentos que el evaluador o el equipo de pruebas opten por realizar de los diez sugeridos por la norma, es decisión propia de cada uno y se tendrán en cuenta factores como la complejidad del software, el nivel de exigencia de las pruebas, el equipo de pruebas, etc. Además podrá optar por unir uno o varios documentos e incluir algún otro si fuere necesario.

Por último, la norma IEE 829 ofrece ocho anexos que son de utilidad para realizar las pruebas, entre los cuales están: bibliografía, ejemplo de esquemas de niveles de integridad, tareas de prueba, ejemplos de documentación, etc.

III. HERRAMIENTA COMPUTACIONAL PARA LA DOCUMENTACIÓN DE PRUEBAS SOFTWARE

Utilizando el paradigma orientado a objetos [10], y siguiendo la metodología ágil de Extreme Programming XP se ha logrado implementar una herramienta computacional para la documentación de pruebas para aplicaciones software desarrolladas en el interior de los grupos de investigación universitarios. Después de un trabajo de investigación y selección en el que se destacan normas tales como IEEE 829, IEEE 1008, BS 7925-1, BS 7925-2 se optó por basar el diseño e implementación de la herramienta a la norma IEEE 829.

La arquitectura general de la herramienta para la documentación de pruebas software está constituida por tres módulos: Módulo de planificación de pruebas, Módulo de desarrollo de las pruebas y el Módulo de reporte de pruebas. La arquitectura general se representa en la figura 4.

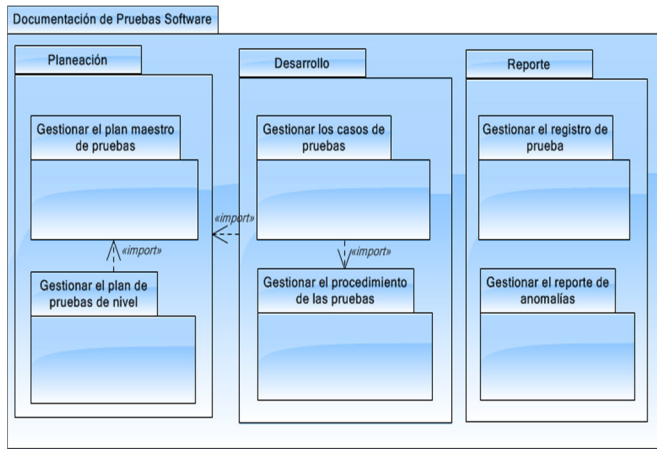


Figura 4. Estructura general módulo documentación de pruebas software

A. Planificación de pruebas

Este módulo consta de dos formularios. El primer formulario es el plan maestro de pruebas que permite registrar datos generales del proyecto, datos tales como: nombre del proyecto, fecha de inicio de las pruebas, estado del desarrollo, versión actual, alcance de las pruebas, una breve descripción del software y referencias externas necesarias para las pruebas o para poder entender el proyecto. Además de estos datos básicos del proyecto, este formulario permite llevar un seguimiento y control sobre los recursos que se disponen en el proyecto como: personal disponible para las pruebas, herramientas utilizadas en el desarrollo y herramientas utilizadas para su realización, técnicas y métodos utilizados, características a probar, un cronograma de las actividades de pruebas y los riesgos que podrían implicar la no corrección de los defectos encontrados. La funcionalidad de este formulario se resume en la figura 5.

Figura 5. Caso de uso plan maestro de pruebas.

En el segundo formulario se separan cada una de las partes o productos del proyecto (código fuente, ejecutable, instalador, etc.) a las cuales se le van a realizar las pruebas. Para cada uno de los productos se documenta: las características a ser probadas con sus respectivas subcaracterísticas, las métricas y los parámetros relacionados con esas métricas. Esto se realiza con el fin de hacer una posterior evaluación de calidad en los productos probados. Además se asocia el personal que participa en el a la planeación y desarrollo de las pruebas de cada producto. En la figura 6 se resume el funcionamiento de este formulario y en la figura 7 se observa el formulario implementado.

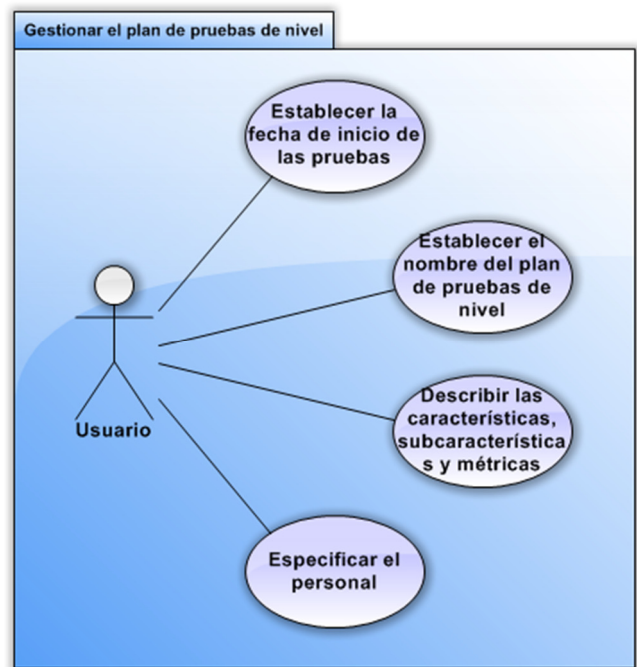
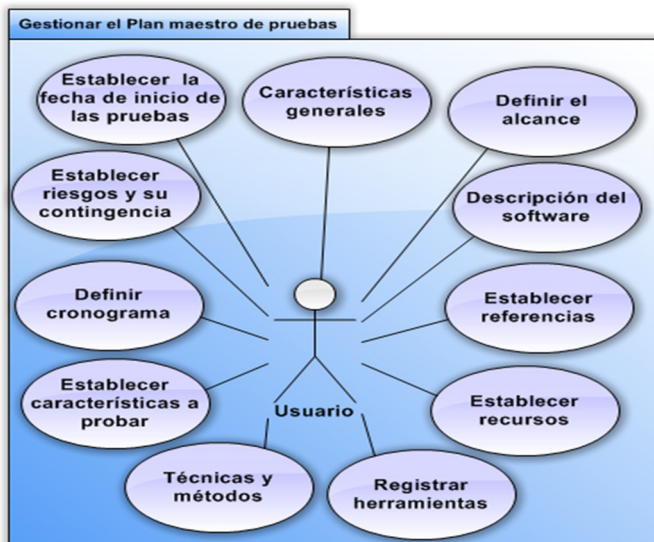


Figura 6. Caso de uso plan de pruebas de nivel.



Características	Subcaracterística	Métrica	Fórmula
Funcionalidad(pruebas)	Implementación de métodos para la entrada de información	Funciones para entrada de datos	NFED
	Implementación de métodos para la salida de información	Funciones para salida de datos	NFSD
Mantenibilidad(pruebas)	Conformidad con el estándar de definición de variables y funciones	Números de Errores en la definición de nombres de variables y funciones	NDED
	Documentación de clases	Documentación de clases	NDCD/NC

Figura 7. Formulario plan de pruebas de nivel.

B. Desarrollo de las pruebas

La función de este módulo es el de documentar los casos de prueba específicos que se realizarán a cada producto, además de un registro general del procedimiento elaborado para las pruebas. Al documentar un caso de prueba, se debe especificar el producto que se está evaluando, el objetivo y una breve descripción del caso, las entradas y salidas que se esperan, describir si este caso de prueba depende de la realización de casos anteriores y finalmente una pequeña descripción de las actividades que se deben realizar en cada caso. Esta funcionalidad se resume en la figura 8.

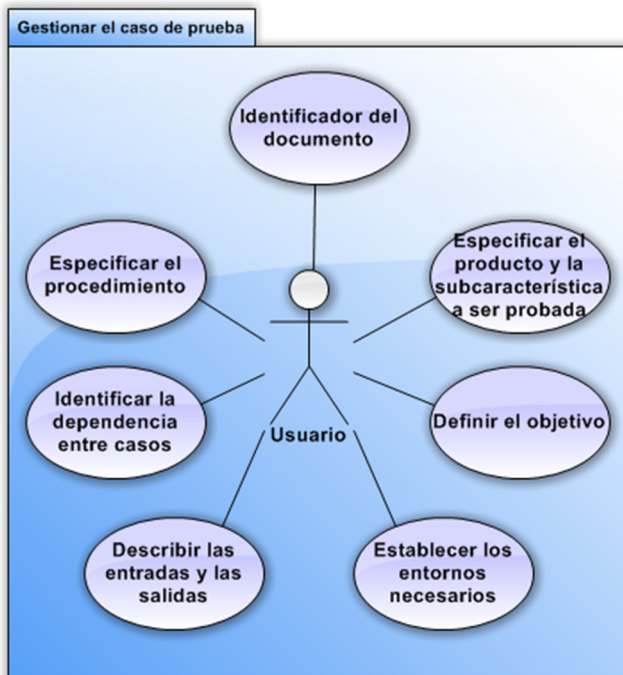


Figura 8. Caso de uso de los casos de prueba.

Al describir el procedimiento general de la pruebas, este se hace secuencialmente; la herramienta se encarga de enumerar automáticamente cada actividad, mientras que el usuario procede a realizar una pequeña descripción de las mismas. En la figura 9 se resume el funcionamiento del módulo y en la en la figura 10 se hace uso del formulario para ejemplificar las actividades que los autores consideran generales en la realización de pruebas software.

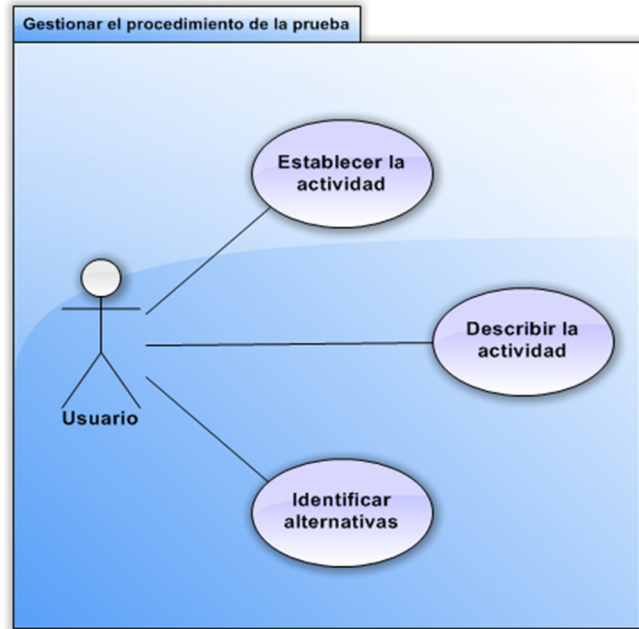


Figura 9. Caso de uso para el procedimiento de prueba

Actividad	Descripción Actividad
1	Planeación de las pruebas software.
2	Diseño de las pruebas software.
3	Especificación de los casos de prueba.
4	Especificación del procedimiento de pruebas.
5	Ejecución de las pruebas.
6	Realizar informes y documentación de los resultados de las pruebas.

Figura 10. Formulario procedimiento general de pruebas.

C. Reporte de pruebas

El objetivo de este módulo es el de llevar un registro de los acontecimientos y descubrimientos significativos del proyecto

en el que se está trabajando, además de un registro de la evaluación de la calidad de los productos a probar. Este módulo se compone de varios formularios los cuales tienen dos tareas fundamentales:

- Llevar un registro de las anomalías encontradas y su proceso de corrección. Esto conlleva a un primer paso, donde el usuario describe detalladamente cada anomalía, documentando su impacto en el software, las posibles causas, las acciones correctivas a aplicar y las implicaciones de corregir o no dicha anomalía. Posteriormente se procede a documentar el proceso de responder el reporte de anomalía, el cual consiste en documentar los trámites, el personal involucrado y los resultados de tratar de solucionar la anomalía. Estos procesos se pueden observar en las figuras 11 y 12. En la figura 13 se puede observar un ejemplo de un reporte de anomalía.



Figura 11. Caso de uso reporte de anomalía.

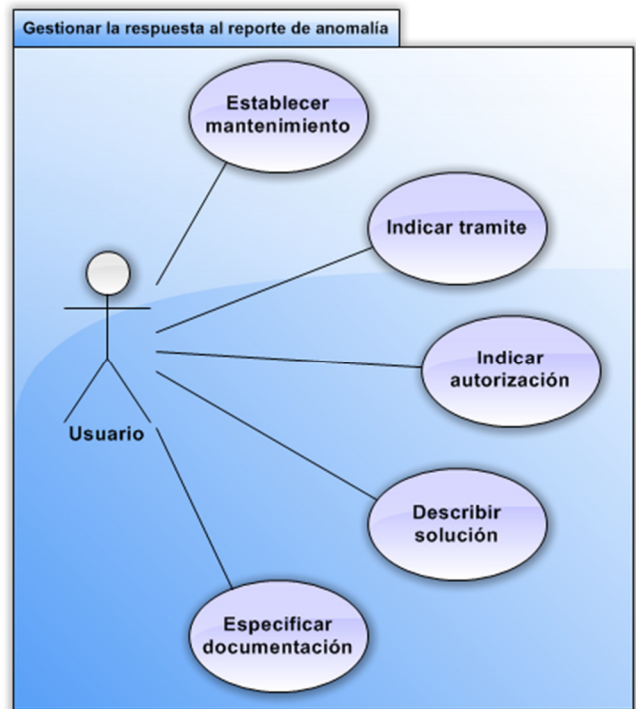


Figura 12. Caso de uso responder reporte de anomalía.

Figura 13. Formulario de reporte de anomalía.

- La segunda tarea fundamental de este módulo es llevar un registro de los acontecimientos significativos que pudieran ocurrir durante la realización de las pruebas. Adicional a esta tarea, en este módulo se pueden registrar las mediciones realizadas a las métricas definidas en un modelo de calidad previamente definido, bien sea creado por el usuario o por un estándar como la ISO 9126[11]. En la figura 14 se resume la tarea planteada.

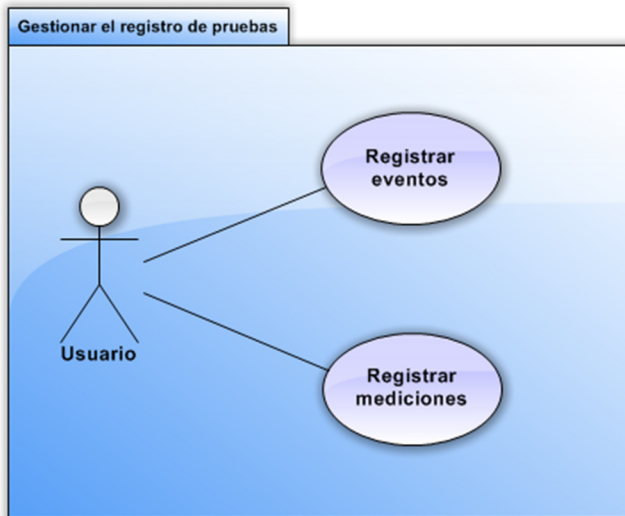


Figura 14. Caso de uso gestionar el registro de las pruebas

D. Reportes

La herramienta tiene un módulo general donde se generan reportes de cada uno de los formularios y de las actividades realizadas, por lo que cualquier operación que el usuario realice en el módulo de documentación de pruebas software también se incluye en la sección de reportes.

IV. CONCLUSIONES

Teniendo en cuenta la importancia de las pruebas se puede resumir como actividades esenciales para su buen desarrollo la planeación, el desarrollo y el reporte de las pruebas. De estas actividades se debe dejar una evidencia como soporte de su realización. La herramienta diseñada fue concebida para facilitar este proceso de documentación de las pruebas en todas sus fases, para lo cual se implementó un conjunto de formularios en Visual Studio .NET™ que permiten la gestión de la información básica para cada una de estas actividades.

La utilización de una norma internacional como la IEEE 829 como base del diseño de la aplicación, permite que los soportes de las pruebas de los desarrollos que se estén realizando tengan concordancia con la documentación exigida internacionalmente, pero que a su vez sea flexible y adaptable a las necesidades y restricciones de desarrollos particulares de los grupos de investigación.

Cabe aclarar que la herramienta no realiza las pruebas automáticamente. Está planteada para ser utilizada como soporte para documentar las pruebas que se pueden realizar utilizando herramientas externas, pero que carecen de un sistema de documentación adecuado.

Por último, es importante señalar que la herramienta se ha empezado a utilizar en la documentación de pruebas de software desarrollado en grupos de investigación de diferentes campos de conocimiento como la biomedicina, los hidrocarburos y los sistemas de información de la Universidad Industrial de Santander.

RECOMENDACIONES

Se recomienda usar la herramienta como soporte a la documentación de pruebas de software desarrollado en otros grupos de trabajo con características similares a los grupos de investigación y con similares restricciones para probar su adaptabilidad.

Los autores expresan sus agradecimientos a la Universidad Industrial de Santander por facilitar la realización del presente trabajo mediante la financiación del proyecto de investigación “MODELO PARA LA DOCUMENTACIÓN DE PRUEBAS DE SOFTWARE INCORPORADO EN LA HERRAMIENTA DE EVALUACIÓN DE CALIDAD DE SOFTWARE DERIVADO DE ACTIVIDADES DE INVESTIGACIÓN” registrado en la dirección de Investigación y extensión de la Facultad de Ingenierías Físico Mecánicas bajo el Código 5563.

REFERENCIAS

- [1]. R. Pressman. *Ingeniería del software-Un enfoque práctico*. Sexta Edición. México. McGraw-Hill 2005. Univ. Maryland, College Park, 1997.
- [2]. *Pruebas de software ¿la salvación, un proceso sin utilidad, trivial, simplemente una moda, o.?* http://www.acis.org.co/fileadmin/Base_de_Conocimiento/XXVII_Salon_Informatica/MariaClaraChoucair-PruebasDeSoftware.pdf [Consultado 27 de diciembre de 2012]
- [3]. BCS. *Standard for Software Component Testing*. British Computer Society. 2001. <http://www.testingstandards.co.uk/Component%20Testing.pdf> [Consultado 01 de mayo de 2013]
- [4]. IEEE. *IEEE Standard for Software and System Test Documentation. Standard IEEE 829-2008*. Software Engineering Technical Committee of the IEEE Computer Society. 2008.
- [5]. ISO. *Software testing-Part 3: Test documentation*. International standard ISO/IEC 29119-3. 2013.
- [6]. IEEE. *IEEE Standard for Software Unit Testing. Standard IEEE 1008-1987*. Software Engineering

Technical Committee of the IEEE Computer Society.
1987.

- [7]. L. Joyanes. *Programación orientada a objetos*. Primera edición. Madrid. McGraw-Hill 1996.
- [8]. ISO. *Software engineering-Product Quality-Part 1:Quality model*. International standard ISO/IEC 9126-1, 2001.