

**AUTOMATIZACIÓN DE PRUEBAS FUNCIONALES, UN
COMPLEMENTO PARA LA CALIDAD DEL SOFTWARE**

**FUNCTIONAL TEST AUTOMATION, A COMPLEMENT TO SOFTWARE
QUALITY**

Lina Paola Colorado Rivera
Ingeniera de sistemas
Estudiante Especialización Gerencia Integral de Proyectos Bogotá,
Colombia.
u1301702@unimilitar.edu.co

Artículo de Investigación

DIRECTOR
Ing. David Alejandro Rincón Castro, M.Sc.



La U
acreditada
para todos

**ESPECIALIZACIÓN EN GERENCIA INTEGRAL DE PROYECTOS
UNIVERSIDAD MILITAR NUEVA GRANADA FACULTAD DE
INGENIERÍA DICIEMBRE DE 2020**

AUTOMATIZACIÓN DE PRUEBAS FUNCIONALES, UN COMPLEMENTO PARA LA CALIDAD DEL SOFTWARE

FUNCTIONAL TEST AUTOMATION, A COMPLEMENT TO SOFTWARE QUALITY

Lina Paola Colorado Rivera
Ingeniera de sistemas
Estudiante Especialización Gerencia Integral de Proyectos Bogotá,
Colombia.
u1301702@unimilitar.edu.co

RESUMEN

El aseguramiento de la calidad del software es una de las etapas del desarrollo de software, las compañías que ofrecen soluciones y servicios integrales basados en tecnología y transformación digital a través de proyectos de software, tienen claro que esta etapa es fundamental para el éxito de los productos que desarrollan y la obtención del retorno de sus inversiones, la calidad no es un proceso fácil y en la mayoría de proyectos se realiza de forma manual por los diferentes QA (Quality Assurance), quienes tienen la responsabilidad de validar el correcto funcionamiento de los artefactos de software que son liberados por los ingenieros de desarrollo. A lo largo del ciclo de desarrollo se ha evidenciado que las pruebas funcionales que son ejecutadas de forma manual, demandan más tiempo del que se había definido en la etapa de planeación, esto ocasiona reprocesos, sobrecostos, incumplimiento de los niveles mínimos de calidad y en ocasiones la pérdida del contrato con los clientes. Teniendo en cuenta lo anterior para esta investigación se realizó una revisión de literatura sobre temas como la automatización de pruebas, metodologías de automatización de pruebas y herramientas de automatización como, Selenium, Katalon Studio, Jmeter y Appium, donde se encontró que la automatización sirve como complemento para la calidad, además hace que las compañías adquieran una ventaja competitiva en la transformación de casos de prueba manuales en script de datos, generando así valor para las empresas en cuanto a costos, calidad del producto y tiempo.

Palabras clave: automatización de pruebas; Calidad software; casos de prueba; transformación digital

ABSTRACT

Software quality assurance is one of the stages of software development, companies that offer solutions and comprehensive services based on technology and digital transformation through software projects, are clear that this stage is essential for success of the products they develop and obtaining a return on their investments, quality is not an easy process and in most projects it is carried out manually by the different QA (Quality Assurance), who have the responsibility of validating the correct operation of software artifacts that are released by development engineers. Throughout the development cycle it has been shown that the functional tests that are executed manually, demanded more time than had been defined in the planning stage, caused reprocessing, cost overruns, non-compliance with the minimum quality levels and sometimes the loss of the contract with customers. Taking into account the above, for this research, a literature review was conducted on topics such as test automation, test automation methodologies, and automation tools such as Selenium, Katalon Studio, Jmeter and Appium, where it was found that automation serves as A complement to quality, it also makes companies gain a competitive advantage in transforming manual test cases into data scripts, thus generating value for companies in terms of costs, product quality and time.

Keywords: test automation; Software quality; test cases; digital transformation

INTRODUCCIÓN

La calidad del software es la combinación de diferentes actividades, tareas, pasos y procedimientos que deben realizar las empresas dedicadas a ofrecer soluciones y servicios basados en tecnología como; consultoría de sistemas, monitorización avanzada, análisis de datos, desarrollo de software a la medida y planes de continuidad del negocio, las cuales se encargan de transformar digitalmente a las organizaciones que existen en sus nichos de mercado, generándoles valor a partir del descubrimiento de oportunidades donde la innovación con tecnología y procesos se conviertan en oportunidades y mejoras del negocio beneficiando así el ecosistema de clientes, proveedores y empleados.

La automatización y digitalización de los procesos en los negocios, constituyen una de las áreas fundamentales de la transformación digital, las empresas dedicadas a esta labor tienen el reto de suplir las necesidades que hoy existen en cuanto a minimizar los tiempos de espera en las entregas de desarrollo de software de calidad, el aprovechamiento al máximo de las nuevas tecnologías, la orientación y capacitación a la alta productividad de los equipos de desarrollo, lo que les facilita llegar mucho más rápido y de manera eficiente a los diferentes mercados.

En la actualidad se evidencia que durante el ciclo de desarrollo de software en la etapa de pruebas funcionales, en algunas empresas dedicadas a esta actividad el personal de aseguramiento de calidad encargado de ejecutar manualmente los casos de prueba no utilizan ninguna herramienta de automatización, lo que ocasiona que deban invertir más tiempo en encontrar errores en el sistema, impactando negativamente los tiempos de entrega que se habían definido en la etapa de planeación, generando retrasos en el cronograma, sobrecostos, incumplimiento de los niveles mínimos de aceptación y calidad, sobrecarga laboral, reprocesos y numerosas pruebas de regresión.

Teniendo en cuenta lo anterior por medio de esta investigación, se busca diseñar un proceso de automatización de pruebas funcionales para el área de calidad que permita complementar las pruebas manuales y así brindar un mayor grado de confiabilidad cuando se apliquen cambios al software y la reducción de los tiempos de pruebas, además de validarse si para todos los proyectos que involucran desarrollo de software son una opción viable, teniendo en cuenta que las pruebas automatizadas pueden ejecutarse una y otra vez luego de ser creadas, son más rápidas que las pruebas manuales y garantizan que lo que no se ha modificado siga funcionando. Para el diseño de este proceso es necesario validar que herramientas existen en el mercado para la automatización de casos de prueba y cuáles pueden ser utilizadas para evaluar resultados de acuerdo a las entradas y salidas del sistema, de tal forma que sean de ayuda para controlar los incidentes que no cumplen con las expectativas del cliente y que pueden restarle valor al producto, además de investigar metodologías y procesos de automatización implementados en compañías que se dedican al negocio de la transformación digital y que pueden ser aplicados para el desarrollo de esta investigación. La automatización de pruebas es una estrategia que agrega competitividad en las organizaciones y ayuda a la simplificación de algunas tareas en la etapa de calidad, en ningún caso debe remplazarse el proceso manual que por décadas se ha utilizado para encontrar errores del sistema, es por esta razón que se debe tener en cuenta el proceso actual que realizan las empresas en cuanto a pruebas manuales y así complementarlo con las pruebas automáticas para mejorar la calidad del software.

MATERIALES Y MÉTODOS

Para realizar esta investigación descriptiva que tiene como objetivo principal diseñar un proceso para el área de calidad de automatización de pruebas funcionales para la ejecución, aseguramiento y control de calidad en las diferentes etapas del desarrollo de software, se realizó una búsqueda en las bases virtuales, ScienceDirect, ProQuest Computing y IEEEXplore pertenecientes a la Universidad Militar Nueva Granada donde se seleccionaron diferentes palabras claves, como “Automatización de pruebas”, “Calidad de Software”, “Aseguramiento y calidad en el desarrollo” la búsqueda de información primero se realizó sin filtros, posterior a eso se realizaron tres filtros de la siguiente manera:

- Por el tipo de documento, texto completo, Rango de años y país.
- Por el tipo de documento, texto completo, rango de años, idioma y países.
- Por el tipo de documento, texto completo, Rango de años.

Adicional se realizó una búsqueda en Google académico con las palabras claves “Aseguramiento y calidad en el desarrollo” inicialmente la búsqueda se realizó sin filtro, luego se realizaron tres filtros de la siguiente manera:

- Ordenando por relevancia
- Ordenando por relevancia y por fecha
- Ordenando por relevancia, por fecha y desmarcando la inclusión de patente.

Para lograr el objetivo principal de esta investigación se definieron los siguientes objetivos específicos:

- 1- Identificar algunas herramientas útiles para la automatización de casos de prueba que sean aplicables a los desarrollos de software.

- 2- Diseñar el paso a paso que se debe llevar para la ejecución de casos de prueba automatizadas para que sea más eficiente el proceso de control de calidad.
- 3- Validar el proceso que actualmente se realiza en algunas organizaciones en cuanto a las pruebas funcionales manuales para el aseguramiento de la calidad del software.

Para el diseño de un proceso de automatización de pruebas lo primero que se debe hacer es responder los siguientes interrogantes:

- ¿Qué beneficios se obtienen de la automatización de pruebas?,
- ¿Cuáles casos de prueba se van a automatizar?
- ¿En qué casos no es recomendable automatizar?

Tener claro que para la automatización de pruebas son indispensables las herramientas de automatización, debido a que la automatización de pruebas se basa en lenguajes de programación para su creación, lo que hace que se conviertan en un esfuerzo de desarrollo de software, en este proceso se está creando un software para probar otro software, por esta razón se deben incluir versiones y planificar mantenimientos para los scripts.

Las herramientas de automatización son aplicaciones independientes o módulos de otras aplicaciones con las cuales se pueden crear los scripts para ejecutar las pruebas, algunas de estas herramientas son Selenium WebDriver, Katalon Studio, Jmeter y Appium.

Selenium es un framework creado para la automatización web que permite ejecutar pruebas automatizadas en diferentes navegadores y sistemas operativos, soporta lenguajes de programación como Java, C#, Ruby, Python, JavaScript y PHP, por otro lado, Katalon studio es una plataforma que está disponible para Windows y Mac, en esta herramienta se pueden automatizar pruebas para web y para Mobile, se pueden automatizar aplicaciones diseñadas en sistemas operativos como Android y iOS.

La herramienta Jmeter permite realizar pruebas de rendimiento en aplicaciones y servicios web, estas pruebas son de gran utilidad para validar la concurrencia de usuarios dentro de una aplicación, como es el comportamiento de una aplicación o un sitio web bajo grandes cargas de trabajo y en qué punto puede ocurrir un posible colapso y por último la herramienta permite monitorear el uso de recursos del servidor.

La herramienta Appium permite automatizar aplicaciones nativas, web e híbridas en dispositivos móviles como Android y iOS, además de ser una herramienta de código abierto.

RESULTADOS Y DISCUSIÓN

A continuación, se relaciona la cantidad de información encontrada en las bases de datos de la universidad Militar Nueva Granada y de Google académico, que fue consultada para diseñar un proceso para el área de calidad de automatización de pruebas funcionales para la ejecución, aseguramiento y control de calidad en las diferentes etapas del desarrollo de software.

Tabla 1: *Resultado de la búsqueda de información.*

| Cantidad de información encontrada | | | | | |
|---|--|-------------------|----------------------|-----------------------|----------------------|
| Fuente de datos | Palabra clave | Sin filtro | Primer filtro | Segundo Filtro | Tercer filtro |
| ScienceDirect | Automatización de pruebas | 218 | 20 | 19 | 51 |
| ProQuest Computing | Calidad de Software | 843 | 761 | 639 | 627 |
| IEEEExplore | Test automation | 1040 | 600 | 385 | 293 |
| Google Académico | Aseguramiento y calidad en el desarrollo | 141 | 127 | 97 | 35 |
| Total | | 1350 | 931 | 755 | 713 |

Respondiendo al interrogante número uno planteado en materiales y métodos, los beneficios que aporta la automatización de pruebas es que se tiene una mayor cobertura de pruebas, ya que permite que se prueben funcionalidades existentes cuando se realizan modificaciones al sistema logrando así, que lo nuevo que se desarrolló tenga un porcentaje menor de pruebas manuales si se tiene en cuenta que los productos software están en continuo crecimiento, lo que imposibilita que manualmente sea alcanzable la cobertura de regresión de las

pruebas, otro de los beneficios es la reducción del tiempo, esto debido a que una vez creadas las pruebas pueden ejecutarse las 24 horas del día por los 7 días de la semana de ser necesario, sin la necesidad de que un QA tenga que estar presente, una vez reducidos los tiempos también se van a ver reducidos los costos a largo plazo, esto teniendo en cuenta que al inicio las pruebas automatizadas requieren un gran esfuerzo, por último la automatización es consistente porque permite ejecutar los mismos casos de prueba siempre siguiendo los mismos pasos.

La respuesta al segundo interrogante es que no todos los casos de prueba del proyecto se deben automatizar y no es viable intentar hacerlo, ya que las pruebas automatizadas no deben sustituir las manuales debido a que estas permiten conocer el funcionamiento del sistema y los pasos que se deben seguir para así poder crear los scripts automatizados. Para este proceso es recomendable automatizar únicamente los casos de prueba que se necesitan ejecutar varias veces y que se convierten en tareas repetitivas y que la mayoría de veces se centran en las áreas de riesgo de la aplicación, además de los casos de prueba que deben ser ejecutados contra diferentes conjuntos de datos o que manejan un alto volumen de datos, las rutas críticas del sistema que son prioridad para el negocio, las funcionalidades que presentan un alto grado de error durante las pruebas manuales y las pruebas que sean necesarias de ejecutar contra múltiples dispositivos, sistemas operativos y navegadores.

Para contestar el tercer y último interrogante no se deben automatizar pruebas cuando las aplicaciones son inestables, esto quiere decir que no se encuentran totalmente desarrolladas o tienen múltiples solicitudes de cambios, no se deben automatizar pruebas que al ejecutarse su resultado no sea predecible, y definitivamente no se debe automatizar cuando no se tenga el tiempo ni los recursos suficientes, esto debido a que la planificación, diseño e implementación de la automatización requiere inicialmente más tiempo y personal capacitado a diferencia de las

pruebas manuales, las pruebas automatizadas son una solución estratégica a largo plazo y no deben ser usadas en el corto plazo para solucionar crisis en los proyecto, no son un remplazo para las pruebas manuales.

Una vez analizada y depurada la información se encontraron algunos conceptos básicos que encierra la automatización de pruebas y la calidad de pruebas:

La calidad se define como el “grado en el que un conjunto de características inherentes de un objeto cumple con los requisitos” (ISO 9000, 2015).

“La calidad del producto software viene determinada por la calidad del proceso con el que se desarrolla. Teniendo un proceso definido y aplicando mejoras sobre este proceso, podremos incrementar la calidad de nuestros productos continuamente”. (Lomprey, 2008, pág. 6)

“Las pruebas funcionales se denominan pruebas funcionales o Functional Testing, a aquellas que tienen por finalidad: 1. Identificar inconsistencias, 2. Asegurar requisitos funcionales, 3. Reducir costos de no conformidades, 4. Evitar reprocesos, 5. Mejorar la productividad, y 6. Aumentar la satisfacción del cliente, (Quality, 2016). Las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creadas, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre esta, el paso siguiente es el pase a producción”. (Morales R., 2017, pág. 141)

Un caso de prueba (test case) es un conjunto de valores de entrada, precondiciones de ejecución, resultados esperados y postcondiciones de ejecución, desarrollados con un objetivo particular o condición de prueba, tal como ejercitar un camino de un programa particular o para verificar que se cumple un requerimiento específico (Institute of Electrical and Electronics

Engineers, 1990).

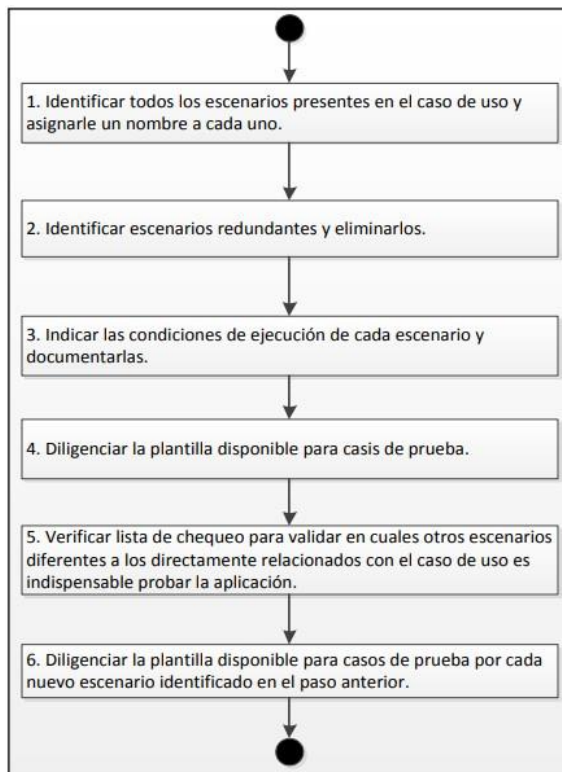


Figura 1: Proceso para creación de los casos de prueba funcionales. Fuente: Tomado de Gonzales (2019).

Propuesta para el diseño de la automatización de pruebas.

La aplicación de estándares de desarrollo y de normas para el software permite lograr calidad técnica y funcional del mismo.

El software es un producto con características muy especiales, hay que tener en cuenta que es un producto que se desarrolla y se centra su diseño, con una existencia lógica de instrucciones sobre un soporte, siendo un producto que no se gasta con el uso como otros y repararlo no significa restaurarlo al estado original, sino corregir algún defecto de origen lo que significa que el producto entregado posee defectos, que podrán ser solucionados en la etapa de mantenimiento. (Piattini, 1996).

“Para realizar la automatización de las pruebas hay diversas herramientas de apoyo así tenemos: las pruebas estáticas, herramientas para la revisión del análisis y el modelado; herramientas para el diseño de las pruebas; herramientas para la ejecución de los casos de prueba y por último las herramientas para la carga y monitoreo”. (Morales R. R., 2017, pág. 145)

Validando las diferentes fuentes se encontró que en el Centro de Ensayos de Software (CES) utilizan un proceso de automatización de pruebas llamado ProTest con herramientas Open Source, en el cual se realizan procesos de testing funcionales apoyados en las herramientas del grupo Selenium (Selenium Core, IDE y Remote Control) (TestNBug in Automation Basics, 2016), para la ejecución de las pruebas, estas herramientas permiten crear y ejecutar pruebas automatizadas sobre aplicativos webs, las cuales pueden ser utilizadas en empresas de software que ofrecen servicios de consultoría, ingeniería de software e implementación de soluciones de negocio.

A continuación, se describe la metodología con los pasos que realiza el Centro de Ensayos de Software (CES) y los cuales pueden llegar a ser apoyo para el diseño del proceso para el área de calidad de automatización de pruebas funcionales para la ejecución, aseguramiento y control de calidad en las diferentes etapas del desarrollo de software.

Pasos metodología ProTest:

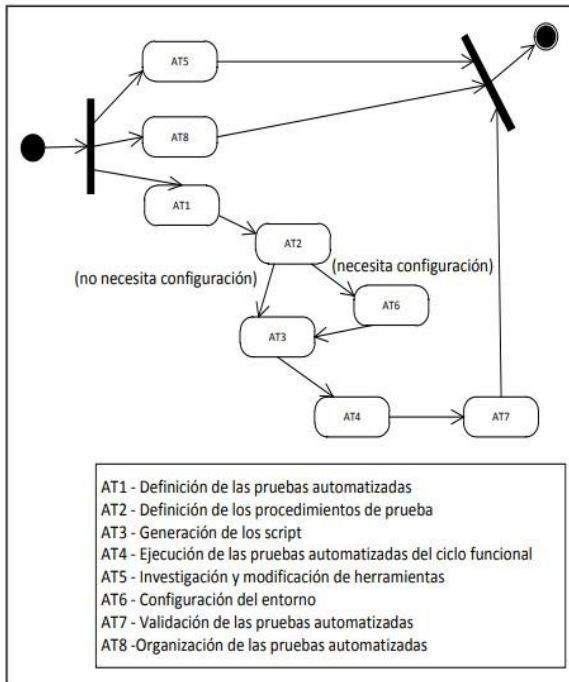


Figura 2: Diagrama de actividades metodológica Centro de Ensayos de Software (CES)
 Fuente: Recuperada de (Morales R., 2017)

Tabla 2: Pasos de la metodología ProTest. (Esmite, 2007, pág. 299)

| Paso | Descripción |
|------------|---|
| AT1 | Definición de las pruebas automatizadas El objetivo de esta actividad es definir el conjunto de ciclos funcionales o funcionalidades que se probarán con las pruebas automatizadas. Esta actividad consiste en definir con el Cliente qué pruebas se van a automatizar (en alto nivel), considerando los ciclos funcionales o funcionalidades que involucran. Para ello se priorizan y evalúan las pruebas y las funcionalidades a probar |
| AT2 | Definición de los Procedimientos de Prueba El objetivo de esta actividad es definir las suites y scripts que conformarán las pruebas automatizadas. A partir de los ciclos funcionales y funcionalidades seleccionados para las pruebas, se especifican las |

| | |
|------------|---|
| | <p>suites y scripts que las ejecutarán. Definir las suites implica definir los scripts que las componen y especificar posibles dependencias de ejecución entre ellas. Para cada script se debe definir la tarea que debe realizar y las verificaciones que debe contener</p> |
| AT3 | <p>Generación de suites y scripts El objetivo de esta actividad es obtener las suites con sus scripts correspondientes. Esta actividad consiste en el armado de cada una de las suites correspondientes al ciclo funcional a automatizar. Se graba o codifica las pruebas obteniendo como resultado los scripts integrantes de las suites. Se verifica el correcto funcionamiento de cada script.</p> |
| AT4 | <p>Ejecución de las pruebas automatizadas del ciclo funcional El objetivo de esta actividad es ejecutar una prueba completa de las suites correspondientes al ciclo funcional y verificar su correcto funcionamiento. Esta actividad consiste en realizar la prueba completa del ciclo funcional en el entorno preparado para dicho fin. En caso de un funcionamiento incorrecto deben realizarse los ajustes necesarios. En esta actividad se verifica el comportamiento de las suites en su conjunto</p> |
| AT5 | <p>Investigación y modificación de herramientas El objetivo de esta actividad es encontrar soluciones a las necesidades que no pueden satisfacerse con las herramientas de automatización que se manejan. Esta actividad incluye recorrer foros, referencias y buscar antecedentes similares a la necesidad planteada. Luego se analizan las posibles soluciones. Estas incluyen instalar nuevas versiones de las herramientas, modificar las herramientas o extenderlas. Esta actividad también incluye búsqueda de nuevas herramientas que asistan en la automatización</p> |

| | |
|------------|---|
| AT6 | Configuración del entorno El objetivo de esta actividad es configurar el entorno de la aplicación que se desea probar para poder ejecutar las suites correctamente y documentar esta configuración. En esta actividad se debe configurar los datos que mantiene la aplicación de manera que permitan la correcta ejecución de las suites. Es importante documentar con el detalle suficiente esta configuración |
| AT7 | Validación de las pruebas automatizadas El objetivo de esta actividad es verificar el correcto comportamiento de los scripts, en el ambiente de prueba del cliente y preparar las suites y scripts generados para la validación del cliente. En esta actividad se prueba y eventualmente se ajustan los scripts para el correcto funcionamiento en el ambiente de pruebas del cliente. El cliente valida comparando las pruebas que espera automatizar con las pruebas que los scripts realizan |
| T8 | Organización de las pruebas automatizadas El objetivo de esta actividad es gestionar los artefactos generados en el proyecto de automatización. Esta actividad consiste en definir, actualizar y ejecutar los procedimientos para la gestión de los documentos, suites y scripts que se generen en el proyecto de automatización |

Teniendo en cuenta los pasos enunciados del proceso de automatización de pruebas ProTest y luego de validar fuentes de información, se puede afirmar que el buen uso de las herramientas de automatización, contar con un proceso estructurado durante el desarrollo de pruebas funcionales y unas eficientes pruebas manuales, son el complemento ideal para realizar una mayor cobertura de pruebas en los diferentes sistemas, asegurando así la de calidad del software, además de brindar beneficios a las compañías como; estructurar las pruebas, mayor detección de defectos, retroalimentación más eficiente al equipo de desarrollo, estabilización

temprana del código fuente, ya que al encontrar errores más rápido el equipo de desarrollo tiene el tiempo para estabilizarlo y por último los resultados que arrojan son confiables. Las pruebas que son automatizadas cuando se ejecutan no se cansan, no tiene prisa y si el contenido de las pruebas no es modificado se obtendrá siempre el mismo resultado; tienen características importantes como su consistencia, la confiabilidad y lo más importante se pueden repetir las veces que sea necesario.

“La popularidad de la automatización de las pruebas también se debe a la aplicación de desarrollo rápido de aplicaciones (RAD) y a su popularidad. Para cumplir la mayor cantidad de actividades relacionadas con las pruebas se deben utilizar herramientas automatizadas, porque a menudo las manuales son laboriosas y propensas a errores, y simplemente no pueden competir con la calidad de la prueba automatizada, especialmente cuando hay que tener en cuenta el cronograma del proyecto”. (Serna, 2019, pág. 171)

Finalmente, al analizar los diferentes autores consultados en esta investigación se evidencia que la automatización de pruebas es una herramienta eficaz para reducir los tiempos y los costos de un plan de pruebas en los proyectos que generen productos software, desde que aparecieron las primeras herramientas de pruebas de automatización son centro de atención por las ventajas que ofrecen en la reducción de errores humanos durante la etapa de pruebas en el ciclo de desarrollo, además de permitir que las personas a través de este proceso adquieran nuevas capacidades técnicas y profesionales, a su vez las empresas adquieran el reto de implementar este proceso en sus compañías de manera eficiente logrando una mejor calidad de los productos software.

CONCLUSIONES

- La automatización de pruebas funcionales combinada con herramientas de software como Selenium, Katalon Studio, Jmeter y Appium ayudan a transformar los casos de prueba manuales en scripts de programación, con el objetivo de controlar la ejecución y comparación de los resultados arrojados en cada prueba, donde lo que se busca es aumentar la efectividad de las pruebas manuales que realizan los diferentes QA.
- Existen muchos retos para la industria del software en cuanto a calidad de sus productos, por esta razón la automatización de pruebas cada vez se consolida más como complemento para las tareas manuales de ejecución de pruebas, convirtiéndose en una estrategia para las compañías dedicadas a esta industria, así como es un reto para los colaboradores de las mismas, ya que deben estar actualizados y capacitados con los cambios de tecnología para que este proceso siga evolucionando y así suplir las necesidades del mercado.
- Teniendo en cuenta que la automatización de pruebas funcionales puede mejorar la calidad del testing, las compañías que diseñen procesos de pruebas automatizadas como el realizado en el Centro de Ensayos de Software (CES) “ProTest” con ayuda de herramientas Open Source pueden llegar a reducir los tiempos y costos de las pruebas manuales.
- Por último, la automatización de pruebas no reemplaza a las pruebas funcionales manuales, para el diseño de esta iniciativa es importante tener claro que estos dos procesos deben mantenerse ya que las pruebas manuales son la base y el complemento de la automatización.

REFERENCIAS BIBLIOGRÁFICAS

- 1- Morales, C; Ruiz, E; Rivera, M; y Ruiz L. (2017) Desarrollo de un modelo de pruebas funcionales de software basado en la herramienta SELENIUM Industrial Data, vol. (20), 139-147.
- 2- IEEE Standard Glossary of Software Engineering Terminology Institute of Electrical and Electronics Engineers, ISBN: 155937067X, 1990.
- 3- Fewster, M & Dorothy Software Test Automation, Effective use of test execution tools. Graham. ISBN 0-201- 33140-3.
- 4- Gonzales, F. (2009) Métodos para generar casos de pruebas funcionales en el desarrollo de software. Revista ingenierías- Universidad de Medellín.
- 5- Esmite, I., Farías, M., Farías, N., & Pérez, B. Automatización y Gestión de las pruebas Funcionales usando Herramientas Open source. (X.C. Computación, Ed.) IV Workshop de ingeniería de Software y Bases de Datos, 294-305.
- 6- Sommerville, I. (2010). Software Engineering. Pearson.
- 7- Valle, K; Delgado, P; Inmaculada, B; y Magallanes, J. Reducción de costes en la Industria 4.0 a través de la prueba del software. Grupo UCASE de Ingeniería del Software. Universidad de Cádiz ,1-4.
- 8- Zapata, C; y Cardona, C. (2011) Comparación de las características de algunas herramientas de software para pruebas de carga. Revista Avances en Sistemas e Informática, Vol. (8), 144-154.
- 9- Mascheroni, A; y Irrazábal, E. (2018) Problemas que afectan a la Calidad de Software en Entrega Continua y Pruebas Continuas. Facultad de Informática. Universidad Nacional de La Plata. La Plata, Buenos Aires, Argentina.; 592-600.

10- Lomprey, G; y Hernandez, S. (2008) La importancia de la calidad en el desarrollo de productos de software. Facultad de Ingeniería y Tecnología Universidad de Montemorelos, México. 1-19.

11- Serna E, Martínez R y Tamayo P (2019) Una revisión a la realidad de la automatización de las pruebas del software. Institución Universitaria de Envigado, Envigado, Antioquia, Colombia Computación y Sistemas, Vol. (23); 169–183.