

Piloto de automatización de pruebas de software en la Unidad de Servicios Tecnológicos del Centro de Servicios y Gestión Empresarial (CESGE) del SENA.

Pilot test of software automation testing in the Unidad de Servicios Tecnológicos of the SENA Centro de Servicios y Gestión Empresarial (CESGE)

Gustavo Adolfo Serna López

. Gestor de Técnico de Laboratorio de Pruebas de Calidad de Software, Servicio Nacional de Aprendizaje (SENA), Medellín, Colombia. sernal@sena.edu.co

Lina María Jaramillo Terán

Gestora de Laboratorio de la Unidad de Servicios Tecnológicos, Servicio Nacional de Aprendizaje (SENA), Medellín, Colombia. ljaramillot@sena.edu.co

Gerardo de Jesús Upegui Londoño

Gestor de Calidad Unidad de Servicios Tecnológicos, Servicio Nacional de Aprendizaje (SENA), Medellín, Colombia, gjupegui@sena.edu.co

Ricardo Gómez Giraldo

Auxiliar Técnico del Laboratorio de Pruebas de Calidad de Software de la Unidad de Servicios Tecnológicos, rigomez@sena.edu.co

Yanyeline Patricia Bueno Loaiza

, Auxiliar Técnico del Laboratorio de Pruebas de Calidad de Software de la Unidad de Servicios Tecnológicos, Servicio Nacional de Aprendizaje (SENA), Medellín, Colombia, ypbueno@sena.edu.co

(Recibido el 02-11-2020. Aprobado el 01-12-2020)

Resumen – La automatización de pruebas de software se fundamenta en el empleo de marcos de trabajo y herramientas que buscan disminuir de forma progresiva la participación de los testers en las actividades que son habituales o iterativas. Este trabajo comprende la implementación de un piloto de automatización de casos de prueba en un proyecto que se desarrolla en la Unidad de Servicios Tecnológicos (UST) del Centro de Servicios y Gestión Empresarial (CESGE) del SENA Regional Antioquia para realizar una comparación entre las técnicas de pruebas automatizadas y pruebas manuales realizadas. Para ello se hizo uso de las herramientas Open Source Cucumber y Gherkin, entre otras.

Palabras clave: Calidad De Software, Pruebas De Software, Automatización De Pruebas De Software.

Abstract – The automation of software tests is based on frameworks and tools that seek to reduce testers' participation in habitual or iterative activities. This work includes the implementation of a test case automation pilot in a project developed in the Unidad de Servicios Tecnológicos (UST) of the Centro de Servicios y Gestión Empresarial (CESGE) of the SENA Regional Antioquia. The project goal is to make a comparison between the techniques of automated tests and manual tests performed. For this, the Open-Source tools Cucumber and Gherkin were used, among others.

Keywords: Software Quality, Software Testing, Software Testing Automation.

Citar como:

G. Serna; L. Jaramillo, G. Upegui; R. Gómez; Y. Bueno. "Piloto de automatización de pruebas de software en la Unidad de Servicios Tecnológicos del Centro de Servicios y Gestión Empresarial (CESGE) del SENA" Revista CINTEX, Vol. 25(2), pp. 45-50. 2020.

1 INTRODUCCIÓN

El concepto de pruebas de software está ligado a una serie de actividades orientadas a establecer herramientas de verificación del funcionamiento previsto de los desarrollos en condiciones deseadas por el cliente, en aspectos técnicos, operativos, funcionales y de resultados, buscando la detección de bugs o errores en el código, combinando la eficiencia y eficacia en el uso de los recursos técnicos y humanos [1], [2].

La fabricación y puesta en funcionamiento de pruebas de software es una labor inherente y complementaria al desarrollo, pues permite certificar que éste marcha correctamente y cumple con estándares mínimos de funcionalidad y operatividad [3]. En este sentido, se debe entender que esta implementación requiere de tiempo adicional al desarrollo, garantizando la identificación de fallas y errores antes de salir a producción, aumentando fiabilidad en su desempeño y brindando certeza de calidad de los productos y servicios ofrecidos [1],[4].

En la dinámica que implica la relación entre desarrolladores y tester, es posible identificar diversos tipos de pruebas de calidad de software: de aceptación, de sistema, de integración, de módulo y unitarias, entre otras. [5, p.], [6]. A su vez hay dos técnicas de pruebas, que son las manuales y las automatizadas. En la Imagen No 1, se pueden apreciar las actividades de desarrollo de software y los niveles de prueba del “Modelo en V” [4].

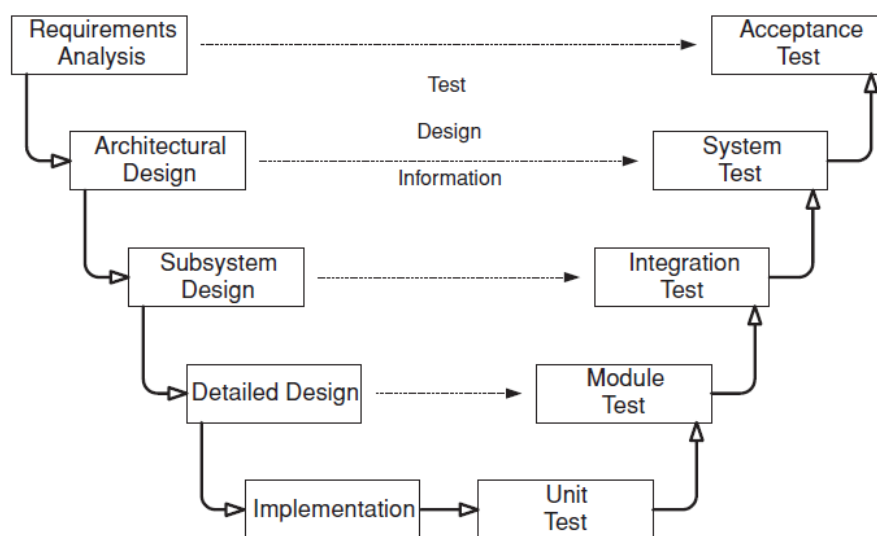


Fig. 1. Actividades de Desarrollo de software y niveles de prueba: el “Modelo V”

La creciente demanda de servicios de desarrollo de software de diferentes niveles de complejidad ha llevado a escenarios en los cuales se requiere de una mayor planeación de recursos humanos para garantizar el cumplimiento de los tiempos acordados [7],[8], [9]. Además, es fundamental realizar una revisión de estrategias encaminadas a agilizar los procedimientos de pruebas funcionales iterativas [10], en donde toma relevancia el concepto de automatización para aminorar la intervención de las personas en las actividades repetitivas, reduciendo el tiempo y esfuerzo requerido y aumentando la cobertura de los casos de pruebas [11], [12].

Dentro de la implementación de estrategias de pruebas en las organizaciones es recomendable integrar el software de automatización como herramienta de apoyo y contribución a la efectividad y calidad; algunos framework son el TestNG y JUnit y dentro de estos, es habitual usar herramientas como SeleniumHQ, Watir, Capibara y Cucumber, las cuales básicamente automatizan los navegadores a través de tareas que los usuarios podrían hacer en aplicaciones web [13], [14], [15] [16]; así como también se encuentra Robot Framework con el cual es posible realizar pruebas de aceptación y desarrollo, [17] y CodeceptJS que es de tipo E2E cuya visión es fundamentada en ambientes y desarrollo por comportamiento (BDD). [8], [18]. Estas herramientas se deben ligar a los lenguajes de programación como: Java, Python y C++ [7], útiles en la codificación para la automatización de pruebas de software.

La Unidad de Servicios Tecnológicos (UST) del CESGE a través del Laboratorio de Pruebas de Calidad de Software realizó un piloto de automatización de casos de prueba a uno de los proyectos que actualmente se desarrollan allí. Se

utilizaron las herramientas Open Source, Cucumber y Gherkin, entre otras, para diseñar dichas pruebas; a su vez, se aplicó la metodología SCRUM donde fue necesario realizar una revisión literaria [19], en la cual se evaluaron marcos de referencia y herramientas para la automatización. Posterior a la selección de los casos más habituales se ejecutaron las pruebas para finalmente realizar una comparación entre las técnicas de pruebas automatizadas y pruebas manuales.

2 METODOLOGIA

Este piloto se realizó sobre uno de los proyectos de desarrollo que actualmente tiene la UST, el cual comprende actividades de servicio y venta, y para el desarrollo de las cuales se seleccionó el módulo de “Ingreso” que tiene las operaciones “login exitoso”, “login fallido” y “recuperar contraseña”.

La investigación se enfocó en el estilo teórico-práctico, por lo cual se realizó una revisión literaria sobre trabajos similares realizados para determinar los framework y las herramientas que podrían facilitar el desarrollo del piloto de automatización, con el fin de realizar la ejecución de pruebas y el comparativo de las características de los resultados de la aplicación de las técnicas de pruebas. Lo anteriormente expuesto se logró mediante el uso de herramientas y artefactos como plan de pruebas, diseño de casos de pruebas, ejecución de pruebas y registro de resultados.

El piloto fue construido bajo el patrón Screenplay (Ver Figura 2), que proporciona diversas ventajas como es el re-uso de pruebas, brindando mayor confiabilidad y comprensión por parte del personal implicado. Para las herramientas seleccionadas, se tuvo en cuenta el aporte de cada una de acuerdo con su funcionalidad en el proceso de la automatización de las pruebas. Se utilizaron las siguientes herramientas: Selenium Web Driver, BDD (Behavioral-Driven Development), Gherkin, Cucumber, Serenity, Gradle.

Finalmente, se procedió con la ejecución de pruebas del aplicativo bajo las técnicas manual y automática de manera paralela, haciendo uso de los casos de prueba previamente diseñados.



Fig. 2. Construcción de Piloto bajo el patrón Screenplay.

3 RESULTADOS

En esta fase se obtuvieron los resultados de la aplicación de las dos técnicas de pruebas manuales (Ver figura 3) y automatizadas (Ver figura 4) al desarrollo seleccionado, con referencia al tiempo y a las actividades involucradas, lo

cual permitió con especial énfasis llegar a conclusiones referentes a las virtudes de la automatización frente a la técnica de pruebas manuales.

El punto de partida experimental consistió en analizar los requerimientos que debe cumplir el desarrollo a probar, el cual aplica para ambas técnicas. Dentro de los valores relevantes del marco de la prueba se encuentran:

- Aplicación de complejidad intermedia.
- Recurso humano: un tester de tiempo completo.
- Módulos probados: 3.
- Periodo de ejecución: 30 días.

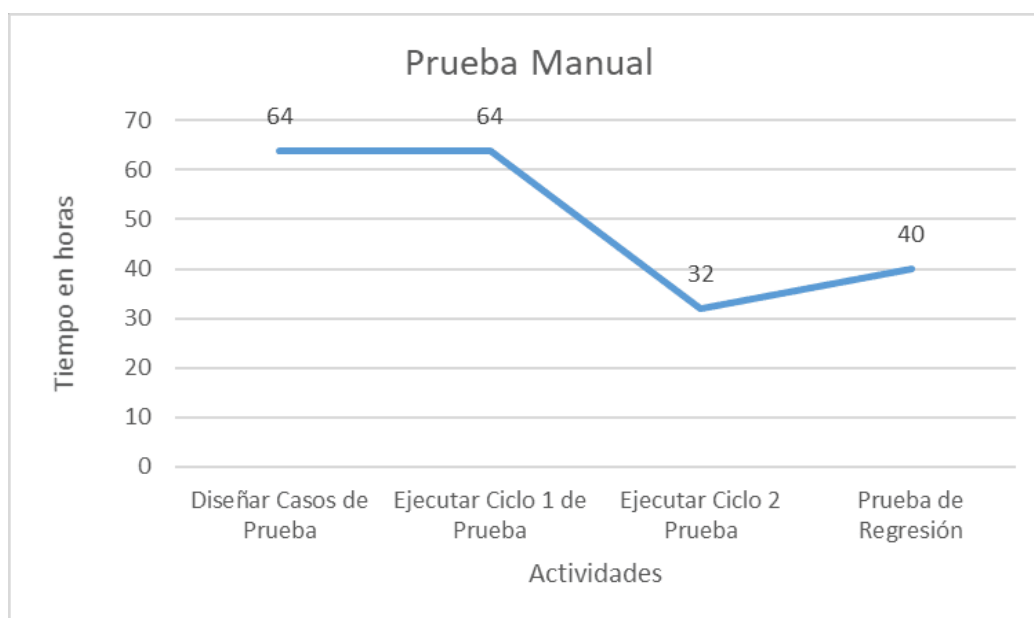


Fig. 3. Resultados pruebas de software manuales.

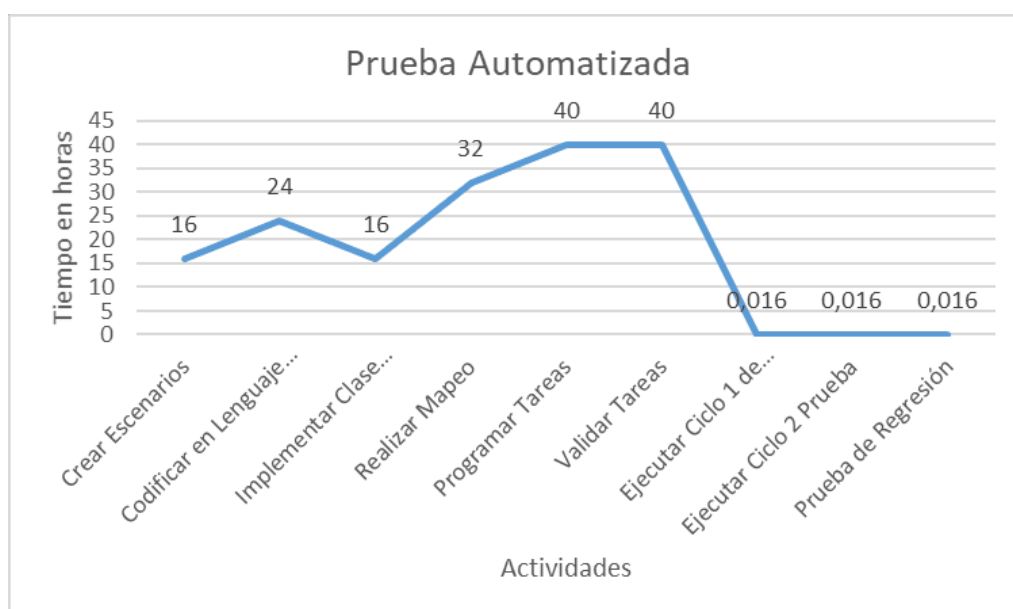


Fig. 4. Resultados pruebas de software automatizadas.

Los resultados arrojaron que la duración en tiempo total para ambas técnicas es diferente. De acuerdo con la Figura 3, el tiempo total manual fue de 200 horas, y según la Figura 4, el tiempo total automatizado fue de 168,03 horas. A su vez, en los resultados se evidencia que en la técnica manual se invierte más horas en la ejecución de las pruebas, mas no en el diseño de estas; como sí se refleja en las pruebas automatizadas, en donde el tiempo de diseño es mucho más alto, lo que se compensa con los tiempos mínimos invertidos en la ejecución de las pruebas.

La ejecución de las pruebas una y otra vez en la técnica automatizada tardan lo mismo, mientras que en la técnica manual es variable por la intervención del factor humano.

Estos escenarios describen un punto de partida, una acción y un resultado esperado apoyándose de expresiones regulares para nombrar algunas variables de entrada que se usarán durante el transcurso de la prueba.

4 CONCLUSIONES

Para la actividad de automatización de pruebas de calidad de software, es imprescindible llevar a cabo una planeación que permita dimensionar las variables y los tiempos que intervienen orientado a cumplir los propósitos iniciales. Esta investigación estuvo enmarcada en realizar la comparación entre la ejecución de pruebas automatizadas y ejecución de pruebas manuales. El resultado obtenido es de tipo prototipo debido a que es una experiencia netamente de laboratorio; sin embargo, hay que tener en cuenta que se efectuó sobre un desarrollo real atendido por el Laboratorio de Pruebas de Software de la Unidad de Servicios Tecnológicos, obteniendo resultados muy cercanos a lo que se esperaba en un ambiente empresarial.

Aplicadas las dos técnicas objeto de esta investigación, pruebas manuales y pruebas automatizadas, y recurriendo a la gráfica de resultados, se logra visualizar las ventajas de las pruebas automatizadas, si bien, en términos de planificación de pruebas, los tiempos invertidos son muy similares, la experiencia de este ejercicio permite concluir que en cuanto al diseño y construcción de los artefactos para las pruebas hay una diferencia de tiempo significativa, reflejado con una alta inversión de tiempo en las pruebas automatizadas; mientras que en la técnica manual, la alta inversión de tiempo se refleja en la ejecución de las pruebas, en este sentido y teniendo en cuenta la reducción del tiempo total de ejecución del proceso, el aumento de la productividad de la UST y la mayor eficacia en la detección de fallas y bugs en tiempo récord, la técnica de automatización demuestra ser inmensamente superior a la técnica de pruebas manuales.

Ante el aumento en la atención de servicios de desarrollo de software de diferentes complejidades, la automatización de pruebas de software se convierte en una prioridad para la UST, y en ese orden de ideas, se deberá fortalecer la investigación y la generación de software a la medida para la automatización de las pruebas, de tal forma que se optimice el recurso para entregar los resultados esperados por las partes interesadas.

Uno de los postulados del SENA apoyado a través de la Unidad de Servicios Tecnológicos es el relacionado con la formación de Aprendices para el trabajo, y precisamente con el Laboratorio de Pruebas, ellos podrán ampliar los conocimientos de los programas de teleinformática teniendo un espacio práctico con proyectos reales enfocados a pruebas automatizadas como hoy lo demanda el mercado.

Finalmente, no se puede desconocer que, la participación del factor humano sigue siendo indispensable para controlar, analizar y mejorar las herramientas diseñadas para lograr mejoras sustanciales en la realización de las pruebas automatizadas. Ha de tenerse presente la premisa que “no todo se puede automatizar” y siempre se requerirá de la intervención de los integrantes del equipo de testing.

REFERENCIAS

- [1] R. F. Catalán San Martín, “Modelo de Desarrollo de Software Guiado por Pruebas y Apoyado con Herramientas de Pruebas de Software,” *Univ. Téc. Federico St. María Dep. Informática*, pp. 1–25.
- [2] R. Gómez Giraldo and L. M. Jaramillo Terán, “Propuesta de laboratorio de certificación en la norma ISO 29119 de pruebas de calidad de software en el Centro de Servicios y Gestión Empresarial del SENA,” *Rev. CINTEX*, vol. 24, no. 2, pp. 46–54, Dec. 2019, doi: 10.33131/24222208.350.

- [3] S. M. Velásquez, J. D. Vahos Montoya, M. E. Gómez Adasme, E. J. Restrepo Zapata, A. A. Pino, and S. Londoño Marín, "Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software," *Rev. CINTEX*, vol. 24, no. 2, pp. 13–23, Dec. 2019, doi: 10.33131/24222208.334.
- [4] P. Ammann and J. Offutt, *Introduction to software testing*, Edition 2. Cambridge, United Kingdom ; New York, NY, USA: Cambridge University Press, 2017.
- [5] M. Callejas-Cuervo, A. C. Alarcón-Aldana, and A. M. Álvarez-Carreño, "Modelos de calidad del software, un estado del arte," *ENTRAMADO*, vol. 13, no. 1, pp. 236–250, 2017, doi: 10.18041/entramado.2017v13n1.25125.
- [6] M. I. Ardila Marín, W. Orozco Murillo, J. Galeano Echeverri, and A. M. Medina Escobar, "Desarrollo de software para la gestión del mantenimiento en los laboratorios de la I.U. Pascual Bravo," *Rev. CINTEX*, vol. 23, no. 1, pp. 43–50, Oct. 2018.
- [7] M. E. Serna, M. R. Martínez, and O. P. A. Tamayo, "Un modelo para determinar la madurez de la automatización de las pruebas del software como área de investigación y desarrollo," *Comput. Sist.*, vol. 21, no. 2, pp. 337–352, 2017, doi: 10.13053/CyS-21-2-2723.
- [8] M. Medina Yacupoma, "Automatización de pruebas para proyectos ágiles aplicando el desarrollo dirigido por comportamiento para una compañía de líneas de belleza," Master thesis, Universidad Tecnológica del Perú, Lima, Perú, 2020. [Online]. Available: <https://repositorio.utp.edu.pe/handle/20.500.12867/3166>
- [9] N. E. Viana-Rúa, L. Y. Arenas-Becerra, and J. A. Patiño-Murillo, "Development of an interactive tool for soft skills enhancement in people with mild cognitive disabilities," *J. Phys. Conf. Ser.*, vol. 1418, p. 012003, Dec. 2019, doi: 10.1088/1742-6596/1418/1/012003.
- [10] C. Adarve Gómez, D. A. Castillo Carvajal, E. J. Restrepo Zapata, and H. Villar-Vega, "A review of virtual reality videogames for job-training applications," *Rev. CINTEX*, vol. 24, no. 1, pp. 64–70, Dec. 2019, doi: 10.33131/24222208.346.
- [11] V. Garousi and M. Felderer, "Worlds Apart: Industrial and Academic Focus Areas in Software Testing," *IEEE Softw.*, vol. 34, no. 5, pp. 38–45, 2017, doi: 10.1109/MS.2017.3641116.
- [12] H. V. Gamido and M. V. Gamido, "Comparative Review of the Features of Automated Software Testing Tools," *Int. J. Electr. Comput. Eng. IJECE*, vol. 9, no. 5, p. 4473, Oct. 2019, doi: 10.11591/ijece.v9i5.pp4473-4478.
- [13] M. Leotta, A. Stocco, F. Ricca, and P. Tonella, "PESTO: Automated migration of DOM-based Web tests towards the visual approach," *Softw. Test. Verification Reliab.*, vol. 28, no. 4, p. e1665, Jun. 2018, doi: 10.1002/stvr.1665.
- [14] P. Mani and M. Prasanna, "Validation of automated test cases with specification path," *J. Stat. Manag. Syst.*, vol. 20, no. 4, pp. 535–542, Jul. 2017, doi: 10.1080/09720510.2017.1395173.
- [15] S. Dhir and D. Kumar, "Automation Software Testing on Web-Based Application," in *Software Engineering*, vol. 731, M. N. Hoda, N. Chauhan, S. M. K. Quadri, and P. R. Srivastava, Eds. Singapore: Springer Singapore, 2019, pp. 691–698. doi: 10.1007/978-981-10-8848-3_67.
- [16] B. García, M. Gallego, F. Gortázar, and M. Muñoz-Organero, "A Survey of the Selenium Ecosystem," *Electronics*, vol. 9, no. 7, p. 1067, Jun. 2020, doi: 10.3390/electronics9071067.
- [17] P. Raulamo-Jurvanen, S. Hosio, and M. V. Mäntylä, "Practitioner Evaluations on Software Testing Tools," in *Proceedings of the Evaluation and Assessment on Software Engineering*, Copenhagen Denmark, Apr. 2019, pp. 57–66. doi: 10.1145/3319008.3319018.
- [18] A. Christy Barus, "The implementation of ATDD and BDD from Testing Perspectives," *J. Phys. Conf. Ser.*, vol. 1175, p. 012112, Mar. 2019, doi: 10.1088/1742-6596/1175/1/012112.
- [19] J. D. Vahos, A. A. Pino, and J. J. Castro Maldonado, "Desarrollo de una herramienta de software para la gestión del mantenimiento de infraestructura en el SENA regional Antioquia," *Rev. CINTEX*, vol. 24, no. 1, pp. 13–19, Dec. 2019, doi: 10.33131/24222208.331.