

Luis Antonio Guevara Aredo , Victor Antonio Villar Garcia, Alberto Mendoza de los Santos.
Universidad Nacional de Trujillo, La Libertad, Perú.

Recibido: 28/02/2022 Revisado: 10/04/2022 Aceptado: 15/05/2022 Publicado: 28/06/2022

Resumen

El control de calidad de un software en la actualidad ha tenido repercusiones en cuanto a las competencias de los usuarios más exigentes y de los servicios o productos que consumen de las empresas. La presente revisión sistemática abarca exactamente los detalles relevantes del desarrollo de pruebas de software en relación al impacto en la calidad del producto. En este contexto, responderemos a la siguiente pregunta: ¿Cómo influye el testeo informático en la calidad de software? Por lo mismo, se hace necesario centrar como objeto de esta investigación, el impacto que generan las pruebas que se realizan en el día a día aplicables en cualquier ámbito, así como los principales aportes que se recapitulan de las diferentes investigaciones en los últimos 7 años. Posterior a la revisión sistemática, se logró identificar que los factores que más influencia tienen respecto a la calidad de un software, son: la elección del modelo de pruebas asociadas al nivel del profesional que las maneja, los procesos de testeo manual y automatizados trabajando en conjunto, la estimación de costos de prueba, la elección de una metodología y el tipo de prueba aplicada.

Palabras Claves: Pruebas de software, calidad de software, testeo informático, desarrollo de pruebas de software.

Abstract:

The quality control of software today has had repercussions in terms of the skills of the most demanding users and the services or products they consume from companies. This systematic review covers exactly the relevant details of software test development in relation to the impact on product quality. In this context, we will answer the following question: How does computer testing influence software quality? For this reason, it is necessary to focus as the object of this

investigation, the impact generated by the tests that are carried out on a day-to-day basis applicable in any field, as well as the main contributions that are recapitulated from the different investigations in the last 7 years. After the systematic review, it was possible to identify that the factors that have the most influence regarding the quality of a software are: the choice of the test model associated with the level of the professional who handles them, the manual and automated testing processes working in set, the estimation of test costs, the choice of a methodology and the type of test applied.

Keywords: Software testing, software quality, computer testing, software test development.

Introducción

Para Jústiz, Gómez y Delgado (2014), la validación de un software como listo para su uso, esta precedida por el esfuerzo de las instituciones en conseguir la calidad requerida. Salomão (2016) explica que el uso intensivo del *testing* asociadas a la especialización de profesionales, es crucial en el desarrollo de software, debido a que permite a los desarrolladores centrarse en la correcta comprensión, modelado y codificación de las reglas de negocio, dejando las etapas de validación a los profesionales de ensayo y calidad. En los términos de Matieli y de Araujo (2020), las fases de prueba tienen los mismos desafíos y problemas relacionados con el costo y el tiempo respecto a las fases de desarrollo. De Pontes (2018) explica que el éxito de una fase de pruebas dependerá principalmente de: especificar requisitos de forma cuantificable, entender a los usuarios del software y sus categorías, enfatizar en pruebas de ciclo rápido y finalmente, crear un software que pueda probarse a sí mismo. Meriem y Abdelaziz (2019) nos dicen que probar un software, es un proceso de verificación y validación para determinar requisitos técnicos y comerciales. Esta actividad siempre ha sido una tarea que consume tiempo y dinero. Afzal, Alone y Glocksien (2016), quienes aportaron información sobre los enfoques de mejora del proceso de prueba de software (STPI), explican que estos marcos guían a entidades desarrolladoras a mejorar sus procesos de prueba. Demostrando que, aunque un marco esté disponible en una organización, no precisamente es aplicable. Lo importante al utilizar cualquiera de los marcos disponibles, es elegir el que mejor se adapte al equipo de trabajo y permita alcanzar los objetivos trazados. Miranda y Gómez (2021), proporcionan información clave sobre una de las pruebas más aplicadas en el *testing* informático, las pruebas de regresión, siendo la “eficiencia” la métrica más usada. Las pruebas basadas en el riesgo se han vuelto bastante populares y se desarrollaron varios enfoques. Sin embargo, los estándares siguen siendo en su mayoría abstractos con respecto a la implementación correcta. Dado que

el estándar internacional recientemente publicado para las pruebas de software, ISO/IEC/IEEE 29119 implica explícitamente los riesgos como parte integral del proceso de pruebas. Großmann, Felderer y Viehmann (2019) reflejan en su investigación la combinación sistemática de evaluación de riesgos y pruebas, es decir, la planificación, el diseño, la implementación, la ejecución y la evaluación de las pruebas. Foidl y Felderer (2018), aseguraron que el testeo informático es una técnica de garantía de calidad esencial para los sistemas modernos de uso intensivo de software que, a menudo, debe realizarse bajo una presión severa, un cronograma desafiante, recursos limitados y la creciente presión de la alta dirección. Además, la prueba completa del software es virtualmente imposible. Como resultado, las pruebas de software efectivas y eficientes deben ser selectivas para asegurar la calidad del producto final. En ese sentido, el estudio se justifica en el impacto que genera la investigación y la revisión de la literatura del testeo informático en la calidad de un producto software, también, busca brindar conocimiento sobre las técnicas que se usan en el mismo, el desarrollo y la priorización de la etapa de prueba de software respecto a su ciclo de vida. Asimismo, la presente revisión sistemática plantea la pregunta: ¿Cómo influye el testeo informático en la calidad de software? Teniendo como objetivo evaluar las principales razones por las que el testeo informático conllevan a un producto software de calidad.

Metodología

La selección de la bibliografía, se realizó teniendo como base la metodología “Prisma”. Para guiar el proceso metodológico se formuló la siguiente pregunta: ¿Cómo afecta el testeo informático a la calidad de un software? Consecuentemente, se realizó la búsqueda y recolección en base a los términos: “software testing” (ST), “computer testing of information systems” (CTIS), “pruebas de software aplicados” (PSA) y “Técnicas y Tipos de Pruebas de software aplicados” (TTPSA). Después se obtuvo respuesta de las siguientes bases de datos: “Scielo”, “Dialnet”, “Redalyc”, “Researchgate”, “Scienccdirect” y repositorios de universidades de Perú, México, Colombia, Cuba, Suecia, USA, entre otros. Se detalla la recolección de 57 artículos iniciales en la *Tabla 1*.

Tabla 1. Tamaño de la muestra

Términos	Springer	Dialnet	Redalyc	Researchgate	Scienccdirect	Otros Repositorios
ST	2	-	2	-	2	-
CTIS	3	3	2	-	-	4
PSA	5	2	4	2	2	-
TTPSA	-	4	3	2	2	2
+	2	1	1	3	-	4

Total: 57	12	10	12	7	6	10
------------------	----	----	----	---	---	----

Observaciones: “+” indica la combinación de término o términos equivalentes al tema de investigación. “-”, hace referencia a 0 artículos encontrados o artículos sin relevancia para este estudio.

El proceso de exclusión de artículos se muestra en la **Figura 1**. La Fase 1 contó con la búsqueda preliminar referente al tema de investigación, además se definieron los términos de búsqueda. La Fase 2 hace referencia a la reducción más significativa del total de artículos encontrados en la Fase 1, por ende, se aplicaron los principales criterios de exclusión, determinando una reducción por artículos iguales, antigüedad e idioma. La Fase 3 corresponde a la depuración final del total de artículos, donde finalmente damos prioridad a los artículos con mayor relevancia, obteniendo un total de 28 artículos listos para el proceso de análisis de información.

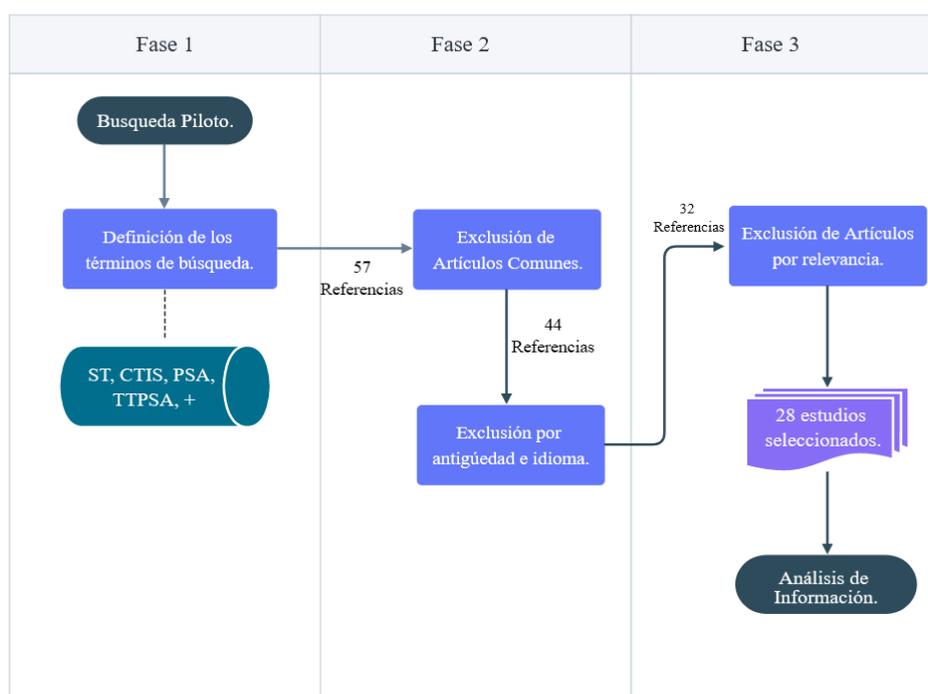


Figura 1. Rondas de exclusión

Fuente: Elaboración propia.

La **Figura 2** muestra la clasificación de los artículos por las regiones de publicación, donde evidenciamos que el 89% corresponden a artículos de investigación publicados en revistas, y el resto estudios recaen en proyectos de grado aplicados, los mismos que fueron considerados por su relevancia y aportes respecto al tema de investigación.

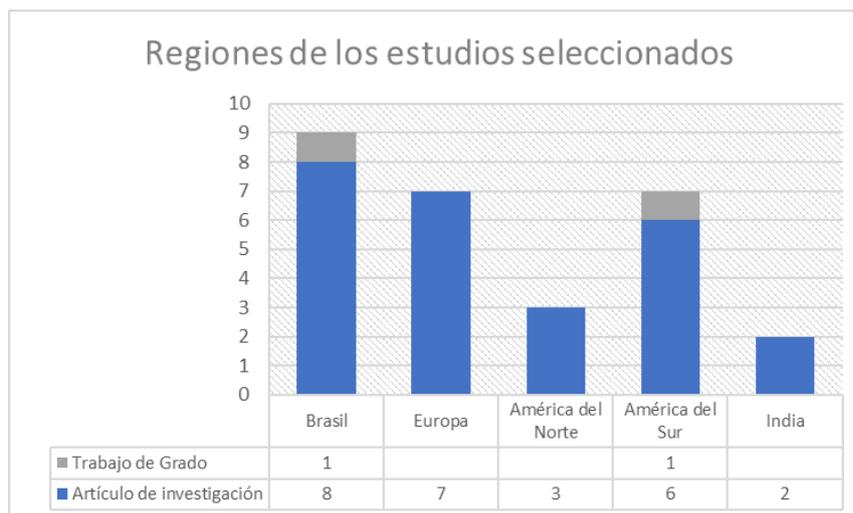


Figura 2. Clasificación de los artículos seleccionados y sus regiones de publicación

Observaciones: Se define Brasil como región independiente de América del Sur, debido a la relevancia que tuvieron sus publicaciones en la selección de artículos para el análisis de información.

Resultados

Tabla 2. Resumen de los aportes relevantes en la literatura abordada.

N°	Autor(es)	Título	Año	País	Aportes Relevantes del Testeo respecto a la Calidad de Software
1	Jústiz Núñez, Dalila; Gómez Suárez, Darlene; Delgado Dapena, Marta Dunia	Proceso de pruebas para productos de software en un laboratorio de calidad	2014	Cuba	La puesta en marcha del proceso de pruebas en los sistemas informáticos de un laboratorio, permitió elevar la calidad del software resultante, así como mejorar la gestión de la institución.
2	Michael Felderer; Rudolf Ramler	Integrating risk-based testing in industrial test processes	2014	Austria	Las pruebas basadas en riesgos optimizan la asignación de recursos, como resultado se obtiene una versión más confiable y optimizada del producto final.
3	Michael Felderer; Rudolf Ramler	Risk Orientation in software testing processes of small and medium enterprises: An E.C.S.	2015	Austria	Las pruebas basadas en el riesgo en las PYME son una parte integral del control de calidad inseparable de otras prácticas, mientras que en las grandes empresas tiene el estatus de una práctica distinta.
4	Charitha Hettiarachchi, Hyunsook Do	Risk-based Test Case Prioritization Using a Fuzzy Expert System	2016	USA	Se mejoró las pruebas de regresión basadas en riesgo, a través del uso de tecnologías difusas.
5	Julián A. Mera Paz	Análisis del proceso de pruebas de calidad de software	2016	Colombia	El testeo en la calidad de software proporcionó sistemas con elevados estándares de calidad y con una reducción importante de fallos.
6	Wasif Afzala; Snehal solob;	Software Test Process Approaches: A Systematic L.R.	2016	Suecia	Muchos de los enfoques del testeo de software no son aplicables a las empresas actuales debido al alto costo que implican.

7	Rudolf Ramler, Michael Felderer	Requirements for integrating Defect Prediction and Risk/Based Testing	2016	Austria	Los requisitos para las pruebas se derivan de la experiencia tanto con la predicción de defectos como con las pruebas basadas en riesgos.
8	Renata Gonzaga Salomão	Análise da Relevância de Testes de Regressão para o Mercado de Desenvolvimento de Software	2016	Brasil	A pesar que las organizaciones consideran a las pruebas de regresión relevantes, no invierten lo suficiente en el área debido a no tener pleno conocimiento de la teoría relacionada con la calidad del software.
9	Fabiane Barreto Vavassori Benitti; Edson Lucas Albano	Teste de Software: ¿o que e como é ensinado?	2016	Brasil	La calidad de un software se ve afectada directamente por las capacidades del <i>tester</i> y su nivel de preparación y especialización.
10	Rudolf Ramler, Michael Felderer, Matthias Leitner	A Lightweight Approach for Estimating Probability in Risk-Based	2017	Austria	Las estimaciones de probabilidad se basaron en números de defectos de la versión anterior. Esto se dedujo a partir de un enfoque ligero (principio del “clima de ayer”).
11	Jiménez Bibián; Oscar Paul	Pruebas de Calidad Aplicadas al Sitio Web Allison	2017	México	Se lograron elevados estándares de calidad en un producto de software basándose en integración de metodologías del proceso de pruebas al ciclo de vida.
12	Adriano Martins de Pontes	Fatores que influenciam a realização de testes em projetos de software: Um estudo qualitativo	2018	Brasil	Se aumentó la capacidad de los <i>testers</i> logrando determinar que, en efecto, el nivel de conocimiento afecta directamente a la calidad del software previa prueba realizada.
13	Harald Foidl; Michael Felderer	Integrating software quality models into risk-based testing	2018	Austria	Mostró que una estrategia de prueba basada en riesgos supera a una línea de estrategia de prueba basada en código de acuerdo con la cantidad de clases que deben probarse para encontrar todos los defectos.
14	Adriano Martins de Pontes; Pablo José L. de Lima; José Adson O. G. da Cunha	Modelagem e testes de software na perspectiva de profissionais da indústria	2018	Brasil	A pesar de la importancia del análisis de modelos y pruebas de softwares, algunos equipos no crean modelos para admitir la codificación, mientras que otros no realizan las pruebas correctamente.
15	Sergio Luis Barbieri	Teste de software na indústria: Um estudo qualitativo	2018	Brasil	Los pasos que busca asegurar la calidad del producto final es la validación, verificación y ensayo (VV&T)
16	Edgar Serna; Raquel Martínez; Paula Tamayo	Una revisión a la realidad de la automatización de las pruebas del software	2019	México	Las pruebas de software durante el desarrollo, mejoró la calidad del producto software, en términos de un menor número de desperfectos presentes en el ciclo de vida.
17	Maria Ines Castiñeira	Automação De Testes De Software: Estudo De Caso Da Empresa Softplan	2019	Brasil	Las pruebas de software automatizadas tuvieron un alto retorno en productividad y agilidad en la ejecución de pruebas a mediano y largo plazo.
18	Jürgen Großmann, Michael Felderer;	A Taxonomy to Asses and Tailor Risk-based Testing in Recent Testing Standards	2019	Alemania	Las taxonomías de las pruebas basadas en riesgos influyen determinantemente en describir sistemáticamente en las áreas de pruebas.

19	Atifi Meriem, Marzak Abdelaziz	A Methodology to do Model-Based Testing using FMEA	2019	India	Se concluye que el enfoque FMEA es capaz de generar casos de prueba priorizados basados en requisitos y modos de falla.
20	Evelyn Joanna Chinarro Morales	Definición e Implementación Del Proceso De P.S. Basado en la NTP-ISO/IEC 12207:2016	2019	Perú	El proceso afectó positivamente en la reducción del tiempo para pruebas funcionales.
21	Gutierrez Zapata Karoly Guadalupe	Implementación De La Herramienta Badboy Para La Mejora De Pruebas Funcionales	2019	Perú	Se redujo el tiempo en desarrollar pruebas funcionales por lo que se consiguió encontrar defectos de software en etapas de pruebas.
22	Thiago Silva-de-Souza; Victor Vidigal Ribeiro; Guilherme Horta Travassos	Estimativa de Esforço em Teste de Software: Modelos, Fatores e Incertezas	2020	Brasil	No se logra un óptimo desarrollo y no se cumplan con las expectativas de calidad. Si no se implementa un modelo de pruebas adecuado.
23	Hema Srikanth, Charitha Hettiarachchi, Hyunsook Do	Requirements Based Test Prioritization Using Risk Factors: An Industrial Study	2020	USA	El enfoque produjo resultados y proporcionaron un esquema de priorización de casos de prueba más efectivo que un enfoque aleatorio.
24	Luciana Vago Matieli; Fernando Oliveira de Araujo	Estimativa De Teste De Software: Levantamento Bibliográfico Em Veículos	2020	Brasil	Una de las mayores dificultades en el alcance de las actividades de pruebas de software es evidenciar claramente las actividades que se realizarán con sus costos y plazos.
25	Omdev Dahiya, Kamna Solanki, Amita Dhankhar	Risk-Based Testing: Identifying, Assessing, Mitigating & Managing Risks Efficiently.	2020	India	Para mitigar riesgos, las pruebas basadas siguen un enfoque en el que el probador no se limita a descubrir las fallas, independientemente de la gravedad y la prioridad.
26	Roberth Melgar Velásquez	Implementación de un modelo según ISTQB para mejorar procesos.	2020	Perú	Se logró detectar la mayor cantidad de defectos posibles y hacer que mayor cantidad de requerimientos puedan ser implementados.
27	Raúl Hernán Rosero; Omar Salvador Gómez	Pruebas de regresión de software: hallazgos iniciales en la industria y academia del Ecuador	2021	Ecuador	La métrica más usada en las pruebas, es la eficiencia en sus tiempos de ejecución.
28	Marina Aparecida da Silva	Teste De Software: Um Estudo De Caso	2021	Brasil	La mayor incidencia de errores se aplica al proceso de desarrollo de software en cuanto a validaciones, errores de código, interpretación de requisitos y análisis de funcionalidad, que son los más comunes que se encuentran en los sistemas desarrollados.

Se encontraron diversos tipos de pruebas que contribuyen a la calidad de software, como lo son las funcionales (interacción usuario y sistema) y no funcionales (experiencia de usuario, diseño, seguridad), encontrando las siguientes consecuencias comunes respecto al tipo de pruebas realizadas:

Tabla 3. Consecuencias en los tipos de pruebas en la literatura abordada.

Prueba (Referencia a Tabla 2)	Tipo de Prueba	Consecuencia Principal
Regresión (6; 8; 16; 27)	Funcional	Disminución de defectos respecto a versiones anteriores.
Aceptación (1; 2; 12; 21)	Funcional	Aumento de requerimientos cumplidos.
Componentes (5; 15; 24)	Funcional	Localización de defectos y comprobación de funcionalidad de módulos.
Estrés (10)	No Funcional	Aumento de la capacidad de usuarios conectados al servidor que soporta al software.
Rendimiento (16)	No Funcional	Aumento en la eficiencia de algunos aspectos específicos del sistema

En la revisión de la literatura, encontramos las pruebas manuales y automatizadas involucradas en la obtención de la calidad de software, en la **Figura 3** se evidencia la comparativa entre países del mundo y la aplicación de *testing* manual y automatizado.

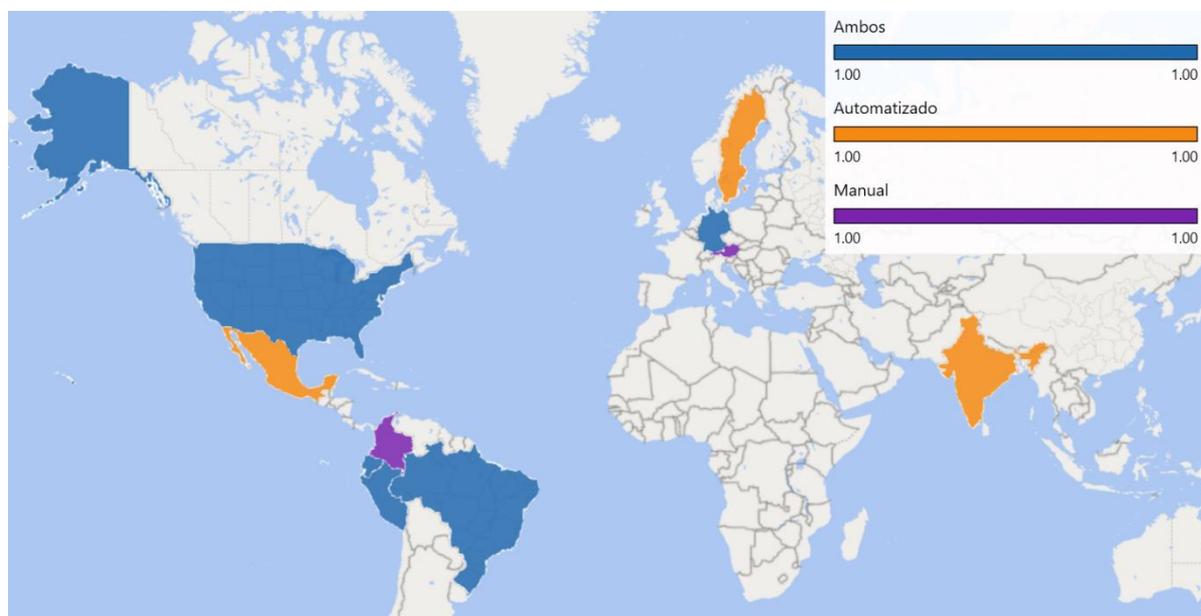


Figura 3. Países de la literatura abordada donde se usan pruebas automatizadas, manuales o ambas.

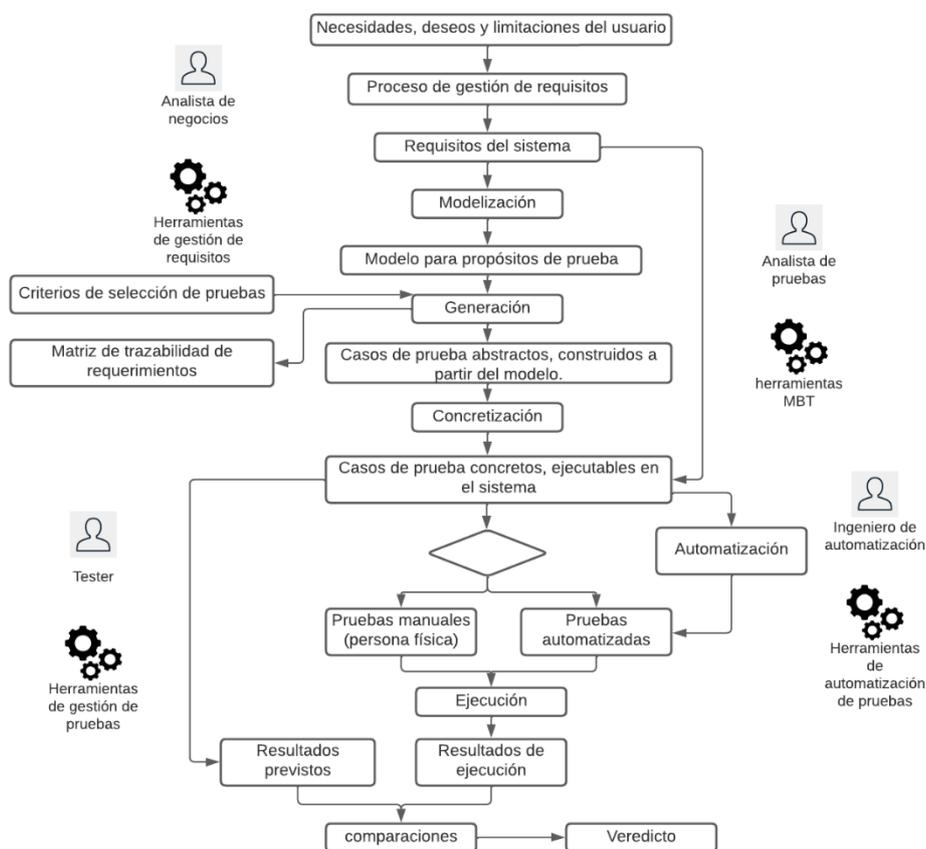
En ese sentido, y considerando los puntos planteados, existen sistemas en los que la única solución para la ejecución de las pruebas es el uso de la automatización de pruebas, teniendo en cuenta el tipo de prueba que se va a realizar y la función a realizar, por lo tanto, haciendo que en ocasiones la ejecución de la prueba manual no sea factible. Entre ellos, se rescatan las herramientas más usadas para pruebas automatizadas, descritos por Castiñeira (2019):

Tabla 4. Herramientas de software más usadas para testing automatizado

Herramienta	Ambiente	Lenguaje
Selenium	Web	Java, Perl, JavaScript, PHP, Python, C#, Ruby
Cucumber	Web	Ruby, Java, NET, Scala y Groovy
TestComplete	Desktop, Mobile, Web	JavaScript, Python, JScript, Delphi, C++ y C#
Telerik TS	Mobile, Web	Angular, Asp.Net, HTML5, JavaScript, Ajax, Silverlight, Ruby, PHP, VB.Net y C#.
Robotium	Mobile	Java
Ranorex	Desktop, Mobile, Web	No usa lenguaje
Appium	Mobile	Java
TestingWhiz	Web, Mobile, BDD	No usa lenguaje

Fuente: Castiñeira (2019)

Como un referente de automatización en la literatura abordada, Meriem y Abdelaziz (2019) nos plantean las bases para los modelos del comportamiento del sistema o su entorno y la generación automatizada de casos de prueba. Esta automatización disminuye el tiempo de prueba y ayuda a lograr una mayor (y medida) cobertura de los posibles escenarios de ejecución.

**Figura 4.** Proceso general de pruebas basadas en modelos

Adaptado de: Meriem y Abdelaziz (2019)

Discusión

Tal y como afirman Paz (2016), si no se identifica un defecto importante y la o las aplicaciones se ejecutan, existen un elevado riesgo de que el software no cumpla con su propósito. A su vez, Afzal, Alone y Glocksien (2016), explican que el estado de las prácticas en las pruebas de software a veces se ignora o se desconoce en las organizaciones ya que las pruebas se realizan de manera ad hoc sin haber designado roles para las pruebas. También, Jiménez Bibián (2017), sostiene que para conseguir la óptima calidad de un producto de software, se debe integrar una metodología de proceso de prueba. Esto implica involucrar las pruebas desde las fases de análisis, pasando por la codificación (pruebas de caja negra o de caja blanca), y así hasta la actualización que permita la adaptación del software a nuevas necesidades del mercado o a errores que no fueron descubiertos en las fases anteriores. Por lo mismo, Barbieri (2018) afirma que se desarrollan técnicas de prueba para verificar y validar requisitos tanto funcionales como no funcionales respecto a la calidad de software. Además, estas técnicas se aplican en las organizaciones a través de estrategias que definen los tipos de pruebas que se realizarán, el orden en que se realizarán, su automatización y la frecuencia de ocurrencia. Para Castiñeira (2019), la automatización está presente en el ámbito del *testing* de software para ayudar a las organizaciones a crecer precisamente en productividad y eficiencia, sin renunciar en modo alguno a la calidad. Pero, Salomão (2016) considera que estas pruebas no pueden automatizarse por completo ya que, en la práctica, es imposible utilizar este tipo de prueba para determinar que un sistema no tiene efectos secundarios no deseados. Contrastando con esto, Castiñeira (2019) afirma que existen dos razones iniciales para automatizar las pruebas de software: El primer argumento está relacionado con el alto crecimiento de los sistemas y aplicaciones, donde se dificulta el control y elaboración de pruebas manuales. La segunda razón se basa en la alta complejidad del software que se está desarrollando actualmente.

Afzal, Alone y Glocksien (2016), afirman que es un hecho bien conocido que las pruebas de software son una actividad que consume muchos recursos. Melgar Velásquez (2020), Barbieri (2018) y Gutiérrez Zapata (2019) coinciden en que las pruebas constituyen más del 30% de los costos generales del desarrollo de software, por lo que optan por las pruebas finales y de rutina, dejando de lado el control específico de calidad. Matieli y de Araujo (2020) afirma que una de las mayores dificultades en el alcance de las actividades de pruebas de software es evidenciar claramente las actividades que se realizarán con sus costos y plazos. Melgar Velásquez (2020) dice que las pruebas funcionales del sistema son importantes porque es donde se realizarán las

operaciones del día a día. Pero Gutierrez Zapata (2019) determinó que el principal problema radicaba en el tiempo que se tomaba en realizar las pruebas funcionales.

Los *testers* están fuertemente motivados por la variedad de trabajos y tareas que requieren creatividad, sin embargo, la categoría “Presión por los plazos” estuvo presente en la mayoría de los resultados obtenidos. Felderer y Ramler (2014) explican que, en muchos dominios de aplicaciones, las pruebas deben realizarse bajo una gran presión debido a los recursos limitados y las limitaciones de tiempo, con la consecuencia de que solo se puede ejecutar un subconjunto de todos los casos de prueba relevantes. Este problema de decisión puede abordarse adecuadamente mediante enfoques de prueba basados en el riesgo para gestionar todas las fases del proceso de prueba. Finalmente, Benitti y Albano (2016) señalan que aunque las pruebas consumen más de la mitad de la vida profesional de un programador, menos del 5% de la educación de un programador se dedica a la actividad de prueba. La falta de actividades prácticas también es un problema identificado en la enseñanza de las pruebas de software en los cursos de pregrado.

Conclusiones

Se logró responder a la pregunta de estudio, determinando muchos de los factores que influyen en la calidad de un producto software respecto a las pruebas realizadas. Por lo mismo, se concluye que:

El uso del modelo de pruebas asociadas al nivel del profesional que las maneja, fue un factor crucial para la obtención de un software de calidad. Además, la estimación de costos de prueba de un sistema respecto a la calidad, se expresó en diversas variables que pueden interferir en el resultado final, tales como: la especialización del profesional de testeo, la naturaleza del software (si las pruebas pueden ser en paralelo), si es posible presupuestar todos los procesos, el ambiente de desarrollo, los requerimientos de los usuarios, la documentación, el tiempo, entre otras.

Las organizaciones consideran la eficiencia en tiempos de ejecución como uno de los principales indicadores para determinar que una prueba es óptima y generará un software de calidad. Seguido de factores como la carga de usuarios y el correcto comportamiento del software frente a la aplicación de una prueba automatizada.

Los procesos de testeo manual y automatizado son muy necesarios en el desarrollo de un software de calidad, disminuyendo estas últimas en gran cantidad el tiempo que un *tester* dedica

a una tarea, reenfocando el tiempo en actividades que nutran a la mejora de las pruebas de calidad. Sin embargo, para pruebas e identificación de defectos específicos, se deberá optar por desarrollo de pruebas manuales.

Como se muestra en el desarrollo de la revisión, las pruebas no corresponden a la última fase del desarrollo de software, y este paradigma representó un limitante en la interpretación de la literatura abordada, redirigiendo nuestro enfoque a cada elemento relevante respecto a todas las etapas del ciclo de vida del software.

Finalmente, recomendamos futuras iniciativas de investigación para informar explícitamente la elección de una metodología y/o tipo de prueba adecuada aplicable al desarrollo de un producto software involucrando pruebas desde las fases de análisis, desarrollo y pruebas.

Referencias

Afzal, W., Alone, S., Glocksien, K., & Torkar, R. (2016). Software test process improvement approaches: A systematic literature review and an industrial case study. *Journal of Systems and Software, 111*, 1-33. <https://doi.org/10.1016/j.jss.2015.08.048>

Barbieri, S. L. (2018). *Teste de software na indústria: Um estudo qualitativo*. <https://doi.org/10.11606/D.100.2018.TDE-12062018-115510>

Benitti, F. B. V., & Albano, E. L. (2016). *Teste de Software: O que e como é ensinado?* 10.

Castiñeira, M. I. (2019). *AUTOMAÇÃO DE TESTES DE SOFTWARE: ESTUDO DE CASO DA EMPRESA SOFTPLAN*. 70.

CHINARRO MORALES, E. J. (2019). *Definición e implementación del proceso de pruebas de software basado en la NTP-ISO/IEC 12207:2016 aplicado a una empresa consultora de software*.

http://cybertesis.unmsm.edu.pe/bitstream/handle/20.500.12672/10587/Chinarro_me.pdf?sequence=1&isAllowed=y

Dahiya, O., Solanki, K., & Dhankhar, A. (2020). *Risk-Based Testing: Identifying, Assessing, Mitigating & Managing Risks Efficiently in Software Testing* [SSRN Scholarly Paper]. <https://papers.ssrn.com/abstract=3565202>

de Pontes, A. M. (2018). *Fatores que influenciam a realização de testes em projetos de software: Um estudo qualitativo*. 13.

Felderer, M., & Ramler, R. (2014). Integrating risk-based testing in industrial test processes. *Software Quality Journal*, 22. <https://doi.org/10.1007/s11219-013-9226-y>

Felderer, M., & Ramler, R. (2015). Risk orientation in software testing processes of small and medium enterprises: An exploratory and comparative study. *Software Quality Journal*, 24. <https://doi.org/10.1007/s11219-015-9289-z>

Foidl, H., & Felderer, M. (2018). Integrating software quality models into risk-based testing. *Software Quality Journal*, 26. <https://doi.org/10.1007/s11219-016-9345-3>

Großmann, J., Felderer, M., Viehmann, J., & Schieferdecker, I. (2019). *A Taxonomy to Assess and Tailor Risk-based Testing in Recent Testing Standards*.

GUTIERREZ ZAPATA, K. G. (2019). *IMPLEMENTACIÓN DE LA HERRAMIENTA BADBOY PARA LA MEJORA DE PRUEBAS FUNCIONALES EN APLICACIONES WEB*. http://repositorio.upci.edu.pe/bitstream/handle/upci/96/T-GUTIERREZ_ZAPATA_KAROLY.pdf?sequence=1&isAllowed=y

Hettiarachchi, C., Do, H., & Choi, B. (2016). Risk-based test case prioritization using a fuzzy expert system. *Information and Software Technology*, 69, 1-15. <https://doi.org/10.1016/j.infsof.2015.08.008>

JIMÉNEZ BIBIÁN, O. P. (2017). *PRUEBAS DE CALIDAD APLICADAS AL SITIO WEB ALLISON*. <https://dspace.colima.tecnm.mx/handle/123456789/725>

Jústiz Núñez, D., Gómez Suárez, D., & Delgado Dapena, M. D. (2014). *Proceso de pruebas para productos de software en un laboratorio de calidad*. Redalyc. <https://www.redalyc.org/pdf/3604/360433597003.pdf>

Matieli, L. V., & de Araujo, F. O. (2020). *ESTIMATIVA DE TESTE DE SOFTWARE: LEVANTAMENTO BIBLIOGRÁFICO EM VEÍCULOS BRASILEIROS E INTERNACIONAIS*. 14.

- Melgar Velásquez, R. (2020). *Implementación de un modelo de gestión de pruebas de software según ISTQB para mejorar el proceso del área de certificación en tecnología web de una entidad financiera*.
https://repositorio.utp.edu.pe/bitstream/handle/20.500.12867/3290/Roberth%20Melgar_Trabajo%20de%20Suficiencia%20Profesional_Titulo%20Profesional_2020.pdf?sequence=1&isAllowed=y
- Meriem, A., & Abdelaziz, M. (2019). A Methodology to do Model-Based Testing using FMEA. *Proceedings of the 2nd International Conference on Networking, Information Systems & Security - NISS19*, 1-11. <https://doi.org/10.1145/3320326.3320342>
- Miranda, R. H. R., & Gómez, O. S. G. (2021). Pruebas de regresión de software: Hallazgos iniciales en la industria y academia del Ecuador. *Polo del Conocimiento: Revista científico - profesional*, 6(11), 1302-1321.
- Paz, J. (2016). Análisis del proceso de pruebas de calidad de software. *Ingeniería solidaria*, 12, 163. <https://doi.org/10.16925/in.v12i20.1482>
- Pontes, A. M. de, Lima, P. J. L. de, & Cunha, J. A. O. G. da. (2018). Modelagem e testes de software na perspectiva de profissionais da indústria: Um estudo qualitativo. *Anais Estendidos do Simpósio Brasileiro de Sistemas de Informação (SBSI)*, 40-43. https://sol.sbc.org.br/index.php/sbsi_estendido/article/view/6204
- Ramler, R., Felderer, M., & Leitner, M. (2017). *A Lightweight Approach for Estimating Probability in Risk-Based Software Testing* (p. 128). https://doi.org/10.1007/978-3-319-57858-3_9
- Salomão, R. G. (2016). *Análise da Relevância de Testes de Regressão para o Mercado de Desenvolvimento de Software do Triângulo Mineiro*. 33.
- Serna M, E., Martínez M, R., Tamayo O, P., Serna M, E., Martínez M, R., & Tamayo O, P. (2019). Una revisión a la realidad de la automatización de las pruebas del software. *Computación y Sistemas*, 23(1), 169-183. <https://doi.org/10.13053/cys-23-1-2782>
- Silva, M. da, Silva, D. da, & Carosia, J. S. (2021). TESTE DE SOFTWARE: UM ESTUDO DE CASO. *Congresso de Tecnologia - Fatec Mococa*, 5(2), Article 2. <https://congresso.fatecmococa.edu.br/index.php/congresso/article/view/290>

Silva-de-Souza, T., Ribeiro, V. V., & Travassos, G. H. (2020). *Estimativa de Esforço em Teste de Software: Modelos, Fatores e Incertezas*. 10.

Srikanth, H., Hettiarachchi, C., & Do, H. (2016). Requirements based test prioritization using risk factors: An industrial study. *Information and Software Technology*, 69, 71-83. <https://doi.org/10.1016/j.infsof.2015.09.002>