



Diagnóstico tecnológico de la pertinencia al implementar un laboratorio de *testing de software*. Caso: Universidad Cooperativa de Colombia, campus Popayán

Technological diagnosis of relevance when implementing a software testing laboratory a case study: "Universidad Cooperativa de Colombia" in Popayán

Julián Andrés Mera Paz¹

Fecha de recepción: 5 de mayo de 2018

Fecha de aceptación: 12 de noviembre de 2018

Cómo citar: Mera P., J.A. Diagnóstico tecnológico de la pertinencia al implementar un laboratorio de *testing de software*. Caso: Universidad Cooperativa de Colombia, campus Popayán. *Tecnura*, 23(59), 68-79. DOI: <https://doi.org/10.14483/22487638.13334>

Resumen

Contexto: Actualmente la Universidad Cooperativa de Colombia, campus Popayán, no cuenta con un espacio o laboratorio de *testing de software*, sin embargo, el *testing* es un elemento fundamental del control de calidad enfocado tanto en el producto como en el proceso. Se plantea un diagnóstico tecnológico con el objeto de validar la pertinencia de su implementación.

Método: En el artículo se realiza una revisión de las definiciones teóricas, las técnicas, los métodos, las herramientas y las características que se sugiere se deben tener en cuenta para la implementación de un laboratorio de *testing* a partir del marco de referencia CMMI y TMMI (referentes de la bibliografía científica). Con el método se enfatiza también la importancia del rol de las instituciones de educación superior y la enseñanza del *testing de software*.

Resultados: El diagnóstico tecnológico es una herramienta que, de forma ágil y eficiente, permite obtener claridad en las necesidades y oportunidades para proponer soluciones reales, como elemento fundamental en la pertinencia y viabilidad de

implementar un espacio para las prácticas y experimentación referente al *testing* y la calidad de software, en la Universidad Cooperativa de Colombia, campus Popayán.

Conclusiones: El diagnóstico tecnológico abre las puertas a profundizar en el contexto del control de calidad y su pertinencia desde los entornos académicos y contribuye a la cohesión entre la industria *software* y el modelo enseñanza/aprendizaje del *software*.

Palabras clave: calidad, laboratorio, prueba, *software*, *testing*, control.

Abstract

Context: Currently the University Universidad Cooperativa de Colombia in Popayán does not have a space or laboratory for software testing; however, the testing is a fundamental element of quality control focused on the product and the process. A technological diagnosis is proposed in order to validate the relevance of its implementation

Method: The article presents a technological diagnosis on the concept, techniques, methods, tools, and characteristics to be taken into account during the

¹ Ingeniero de Sistemas, especialista en Redes y Servicios Telemáticos, magíster en Dirección Estratégica de Telecomunicaciones. Estudiante de doctorado en proyectos TIC. Docente de la Universidad Cooperativa de Colombia. Popayán, Colombia.
Contacto: julian.mera@campusucc.edu.co

implementation of a testing laboratory, based on the CMMI and TMMI reference framework. The research takes as reference the scientific bibliography, which focuses on the role of higher education institutions and the teaching of software testing.

Results: The technological diagnosis is a tool that, in an agile, efficient and effective way, allows to obtain clarity in the needs and opportunities, to propose real solutions, as a fundamental element in the pertinence and viability of implementing a space

for practices and experimentation regarding software testing and quality, at the University Universidad Cooperativa de Colombia, in Popayán.

Conclusions: The technological diagnosis opens the doors to deepen the context of quality control and its relevance from academic environments, contributing cohesion between the software industry and the teaching-learning model of software.

Keywords: quality, laboratory, test, software, testing, control.

INTRODUCCIÓN

El control de calidad, visto como la investigación enfocada en el producto, y como el proceso que brinda información relativa del *software*, se convierte en un marco de referencia que permite medir la combinación de características y atributos con los requerimientos funcionales, no funcionales y deseados por el usuario, para garantizar la calidad (Wong, 2016). A partir de la anterior definición y de que los últimos años la tendencia del desarrollo tecnológico está avanzando con alta velocidad, en dispositivos electrónicos, *big data*, blockchain, internet of things, human computer interaction, serious games, etc., han conllevado a que los productos y procesos software se ajusten a esa alta velocidad de desarrollo, donde las empresas, profesionales, estudiantes e interesados, se engranan a una producción masiva de hardware y software cada vez más compleja, de igual manera para la producción y desarrollo se exige de espacios e infraestructura, técnicas, métodos, herramientas y características más robustas y acompañadas de modelos de enseñanza-aprendizaje que satisfagan el avance tecnológico globalizado. En un escenario académico como la Universidad Cooperativa de Colombia, campus Popayán, la producción y tratamiento de los procesos y productos *software* se tornan más complejos, por su condición de universidad multicampus y enfocada en la economía solidaria (UCC, 2013), ya que

el papel protagónico lo desempeñan estudiantes, docentes y egresados que pretenden dar soluciones a diferentes comunidades en contextos cambiantes. Por esta argumentación es clave realizar un diagnóstico tecnológico que permita analizar la pertinencia al implementar un laboratorio de *testing de software*.

Se establece la fase de análisis de literatura científica, donde se apropia como “una síntesis de trabajo campo que respalda la identificación de preguntas de investigación específicas” (Rowley y Slack, 2004). El artículo se caracteriza por la categorización de documentos relevantes y significativos que analizan la información de teorías, metodologías y prácticas relacionadas con el *testing de software*, desde una perspectiva metodológica de lo clásico, evolutivo y de agilidad del software, para la implementación de un laboratorio. El análisis se caracteriza por seguir una metodología estricta y organizada, tomando como referente a Vom Brocke (2015), quienes postulan que las fases adecuadas de la revisión literaria son: alcance, conceptualización, búsqueda de literatura, análisis y síntesis.

En la fase de revisión del contexto académico, se identificaron las instituciones académicas, que han tenido relación con la implementación de laboratorios y *testing de software* en un contexto internacional, nacional y local.

En la sección “Discusión”, se argumentan los pro y contras de implementar un laboratorio de

testing de software, los cuales se contrastan con los indicadores cuantitativos y cualitativos de entornos académicos.

En los resultados, se fundamenta de forma estructurada lo más significativo del diagnóstico tecnológico, con una secuencia lógica donde de forma concisa, se confirma la pertinencia o no de implementar un laboratorio de *testing de software* para la Universidad Cooperativa de Colombia, campus Popayán.

En la fase de conclusiones, se establece la importancia de los procesos de investigación referentes al *testing* y la calidad de *software* aplicables a laboratorios de instituciones académicas.

METODOLOGÍA

Para este proceso se asumen los aspectos teóricos y prácticos del *testing* y la aplicación a laboratorios en entornos universitarios. La pregunta central fue: ¿Cuál es la pertinencia de implementar un laboratorio de *testing de software* en un entorno universitario? Para responder, se parametriza la información inicialmente en los conceptos comunes, bajo una exploración inicial de la temática donde los autores parten de una posición neutral. Además, se pretende conceptualizar desde el interés del público en general. Para clarificar las características y condiciones se presentan en la tabla 1 (Cooper, 1988).

Tabla 1. Características y condiciones del alcance del análisis de literatura científica

Características		Condiciones seleccionadas para el análisis	
Enfoque	Resultados de investigación	Procesos de investigación	Prácticas y estudios
Objeto	Integración	Crítico	Situaciones específicas y relacionados con el tema central
Organización	Cronológica (últimos 5 años)	Conceptual	Metodologías aplicadas
Perspectiva	Neutral	Neutral	Propositivas
Entrevista	Científica	Habitual	Habitual
Comprensión	Total	Discriminativa	Representativa

Fuente: adaptado de Cooper (1988).

Conceptualización

El *testing* se ha asumido como una actividad en la que solo se ejecutan unos casos de prueba y como una función que la puede realizar cualquier persona. Esta afirmación es falsa, ya que el *testing* es un proceso que incluye una gama de actividades, fases y funciones antes, durante y después de la ejecución de pruebas; además, la percepción puede cambiar con los objetivos que se vayan a definir para la ejecución de las pruebas (Lackner y Schlingloff, 2017).

La norma de International Software Testing Qualifications Board (2016) define el *testing* como

[...] el proceso que sigue paso a paso las fases del ciclo de vida, teniendo en cuenta las variables

estáticas y dinámicas, relacionadas con la planificación, preparación y evaluación de productos *software* y productos relacionados, para determinar que cumplen los requisitos especificados, para demostrar que son aptos para el propósito y para detectar defectos. (p. 13)

Al hablar de control de calidad con énfasis en *software*, es necesario mencionar el modelo de madurez de pruebas integrado (TMMI por sus siglas en inglés), ya que brinda de manera estructural y organizativa las pautas para mejorar en los procesos de pruebas y control de calidad de *software*. Se caracteriza porque las entidades que utilicen este marco de referencia, a medida que van alcanzando los objetivos, se van certificando en los cinco niveles establecidos en el modelo.

Se observa entonces que el *testing* es un proceso más amplio y su correspondencia con el control de calidad en *software* es correlacional, por tanto, es estratégico interpretar su concepto.

La calidad de *software* en general tiene múltiples definiciones y puntos de vista que conllevan a percepciones diferentes por parte de sus técnicos, tecnólogos, profesionales y demás usuarios, pero se inclinan en puntos en común, que permiten articular marcos de referencia o guías para la aplicación o tratamiento de la calidad. Algunas de las definiciones son:

- Según (Callejas, 2017), “la calidad de *software* es el grado en el que el sistema, componente, modulo o proceso cumple con los requisitos específicos, necesidades y expectativas del cliente o usuario” (p. 238)
- Por su parte, según la norma (ISO 8402, 1994), la “calidad de *software* es la totalidad de propiedades y características de un proceso o servicio que le confiere su aptitud para satisfacer unas necesidades expresadas o implícitas” (p. 75).
- Para (Pressman, 2010),
la meta del aseguramiento de la calidad es proveer al equipo administrativo y técnico los datos necesarios para mantenerlo informado

sobre la calidad del producto, con lo que obtiene perspectiva y confianza en que las acciones necesarias para lograr la calidad del producto funcionan. Por supuesto, si los datos provistos a través del aseguramiento de la calidad identifican los problemas, es responsabilidad de la administración enfrentarlos y aplicar los recursos necesarios para resolver los correspondientes a la calidad. (p,351)

Estas tres definiciones brindan pautas para asegurar que la calidad de *software* se debe relacionar con el producto y con el proceso, donde factores como los requerimientos funcionales y no funcionales, además de los deseados por el usuario, deben ser componentes medibles dentro de las dimensiones de calidad (tabla 2).

Por otra parte, el concepto de *laboratorios* en Colombia toma fortaleza en el respaldo estatal. El artículo 71 de la Constitución Política de Colombia señala: “El Estado creará incentivos para personas e instituciones que desarrollen y fomenten la ciencia y la tecnología, las demás manifestaciones culturales y ofrecerá estímulos a las personas e instituciones que ejerzan estas actividades”. En el artículo 69: “se garantiza la autonomía universitaria. Las universidades podrán darse sus directivas

Tabla 2. Dimensiones de calidad de *software*

Dimensión	Descripción
Calidad del desempeño	Se refiere al contexto, las actividades y particularidades planteadas en el modelo de requerimientos.
Calidad de las características	Genera sorpresa y agrado en la primera impresión del usuario.
Confiabilidad	El <i>software</i> tenga todas las características de funcionalidad y capacidades en el momento necesario, sin alteraciones y sin incidentes.
Conformidad	El <i>software</i> siga y cumpla con los estándares, las normas y regulaciones locales e internacionales
Durabilidad	El <i>software</i> se adapta a los cambios y facilita ajustes o mejoras.
Servicio	El mantenimiento y la depuración se pueden hacer en un tiempo breve aceptable.
Estética	El <i>software</i> visualmente se adecua a estándares aceptables por el usuario, haciendo que las actividades en él se observen de forma adecuada.
Percepción	Recibe en general buenos comentarios por parte de los usuarios.

Fuente: Pressman (2010).

y regirse por sus propios, estatutos, de acuerdo con la ley". "El Estado fortalecerá la investigación científica en las universidades oficiales y privadas, y ofrecerá las condiciones especiales para su desarrollo".

La Universidad Cooperativa de Colombia ha estructurado en sus campus los ambientes prácticos de aprendizaje (APA), unos espacios físicos o virtuales diseñados para que de manera experimental y controlada el estudiante confronte, aplique

o genere conocimiento en un escenario similar o modelado a su ejercicio profesional. Entre estos ambientes se encuentran los laboratorios, simuladores, software y modelos a escala (UCC, 2014). En base a lo establecido en las normas internas de la Universidad cooperativa de Colombia, se plantea la tabla 3 de variables acorde a la normativa, con ella se refuerza la pertinencia de implementar un laboratorio de testing de software en el campus Popayán.

Tabla 3. Tabla de variables acorde a la normativa

Variable	Tipo de variable	Definición de la variable	Indicador
Constitución Política	Cualitativa, independiente	Máxima norma de acción legal y jurisprudencial en el Estado colombiano	Cumple
Sistema de habilitación institucional	Cualitativa, independiente	Grupo de reglas, requisitos e instrucciones, establecidas en el interior de la Universidad Cooperativa de Colombia, para registrar, verificar, monitorear y efectuar los requerimientos básicos de capacidad tecnológica e investigativa, bajo parámetros administrativos internos.	Cumple
Infraestructura física	Cualitativa, dependiente	Es el espacio físico para la implementación de un ambiente práctico de aprendizaje, donde puedan confluir los elementos, herramientas, tecnologías y demás elementos necesarios para que pueda funcionar en óptimas condiciones y ofertar servicios o productos a clientes internos o externos.	Cumple
Ambientes virtuales para la aplicación de <i>testing</i> a partir de soluciones tipo <i>cloud</i>	Cualitativa, independiente	Es el espacio virtual donde a través del aprovechamiento de soluciones tipo <i>cloud</i> se pueden diseñar y crear diferentes tipos de pruebas.	Cumple

Fuente: elaboración propia.

BÚSQUEDA DE LA LITERATURA

Con base en la información anterior se decide seleccionar: Google Scholar, Springer, Engineering Village, Scopus, IEEE Explorer, Proquest, congresos de aseguramiento de calidad realizados por comunidades europeas y latinoamericanas, con el objeto de profundizar en la temática *testing*, laboratorio y la relación entre ellos.

Se evidencia en primera instancia que la combinación específica de búsqueda "laboratory + testing + software" son inexactas o muestran a una o

dos de las palabras combinadas con otras como "internet of things laboratory", "mobile application testing", "The Oracle problema in software testing", "Imaging analyzer for testing", "A method of testing software", entre otros.

De las búsquedas se encuentran aspectos comunes o mencionados, que se aplican en el *testing* de *software* y que son soporte para que desde este diagnóstico tecnológico se pueda validar la implementación de un laboratorio de *testing* de *software* (tabla 4).

Tabla 4. Factores comunes de *testing* de *software* en búsqueda literaria

Factores	Software de escritorio	Aplicaciones, sitios o páginas web	Aplicaciones móviles
Usabilidad	Aplica en bajo grado	Aplica en alto grado	Aplica en alto grado
Robustez	Aplica en alto grado	Aplica en alto grado	Aplica en alto grado
Portabilidad	Aplica en bajo grado	Aplica en alto grado	Aplica en alto grado
Interacción con humano	Aplica en bajo grado	Aplica en alto grado	Aplica en alto grado
Efectividad	Aplica en alto grado	Aplica en alto grado	Aplica en alto grado
Eficiencia	Aplica en alto grado	Aplica en alto grado	Aplica en alto grado
Satisfacción de usuario	Aplica en alto grado	Aplica en alto grado	Aplica en alto grado
Accesibilidad	Aplica en alto grado	Aplica en alto grado	Aplica en alto grado
Seguridad	Aplica en alto grado	Aplica en alto grado	Aplica en alto grado

Fuente: elaboración propia.

Para una mejor comprensión, se brindan las definiciones de los factores:

Usabilidad: hace referencia al grado de utilización del *software* para cumplir con los objetivos específicos planteados por el usuario.

Robustez: es la solidez, estabilidad y firmeza del *software*.

Portabilidad: característica de acople y ejecución de un *software* en diferentes plataformas.

Interacción con humano: capacidad del *software* para hacer intuitivo su uso y comprensible al ser humano.

Efectividad: cumplimiento de las tareas o acciones por parte del *software* en un tiempo específico y con un alto grado de éxito.

Eficiencia: es el menor grado de esfuerzo y utilización de recursos de *software* y *hardware* para cumplir con las tareas, acciones o instrucciones orientadas por el humano.

Satisfacción del usuario: mide el cumplimiento de los requerimientos y los sentimientos de agrado del usuario, cuando el *software* culmina las acciones, tareas o instrucciones orientadas por él.

Accesibilidad: grado de utilización del *software* o sus servicios, independiente de las capacidades técnicas, cognitivas o físicas.

Seguridad: capacidad de proteger la consistencia y reserva de la información creada o almacenada en un *software*.

En base a esta búsqueda literaria, se observa que para la aplicación de pruebas se han investigado y desarrollado varios modelos para *testing*. Entre ellos: TestPAI (Sanz, 2008); TMMI (Van Veenendaal y Wells, 2012); MPT.BR, un modelo brasileño de madurez para pruebas que se presenta como una alternativa para las empresas que buscan aumentar la calidad de sus productos y el retorno de inversión (Furtado, 2012); TPI, tomado como una guía que contribuye a la práctica disciplinada de pruebas de *software* y se permite agregar valor a procesos y productos (García, 2014); TPI Next, el modelo de mejora del proceso de pruebas dirigido por negocios ha demostrado ser una de las líderes para evaluar la madurez de un proceso de pruebas de una organización, se destaca por medir las áreas clave, niveles de madurez, puntos de control, sugerencias de mejora y mejora paso a paso (Ewijk, 2013).

Se presenta avances en la prueba de líneas de productos *software*, donde surgen varias dudas sobre los modelos y garantía de la calidad de *software* (Lackner, 2017) plantea un análisis desde un punto de vista de ingeniería de *software* empírico, que busca desde la aplicación de métodos, herramientas, modelos y algoritmos.

Por su parte, la norma ISO/IEC 29119 (IEEE 1012 e ISO, 2013) establece una guía para la fundamentación y desarrollo del *testing* en diferentes

entornos, que se está perfeccionando y actualizando por la organización de estándares ISO.

Análisis y síntesis de la búsqueda literaria

Se observa que la relación entre “laboratorio + testing + software” es aislada; los proyectos de investigación no han centrado su esfuerzo o relevancia en esta unión de conceptos, procesos, métodos y procesos de enseñanza/aprendizaje. Los programas académicos de universidades con enfoque a desarrollo *software*, según la búsqueda literaria, en su etapa de formación no cuentan con procesos o espacios donde se hayan implementado laboratorios de *testing de software*. Esto es un factor que puede incidir en que los egresados de esos programas desconozcan prácticas, técnicas o modelos de *testing*, que influyan en la calidad de *software*.

Con los resultados de la revisión literaria, se puede observar que el *testing* y la calidad de *software* son necesarios e importantes para la industria, por ello se han desarrollado técnicas, modelos y otros que han sido implementados y tienen un grado de confiabilidad alto. Esta relación de la industria y el *testing de software* se fortalecería, si los estudiantes u egresados en su etapa de formación tienen el acercamiento a un laboratorio de *testing de software* y pueden aplicar marcos de referencia, que les permitan elaborar informes del estado de un *software* y su comportamiento en un ambiente simulado, esto conlleva a una ventaja que consolidaría los procesos de desarrollo de *software*, incrementando el grado de los factores de *testing de software*.

REVISIÓN DE CONTEXTO ACADÉMICO

En el contexto internacional se mencionan los casos más relevantes en cuanto al acercamiento a laboratorios de *testing* en el contexto académico e industrial:

En Estados Unidos, la Organización Internacional de Normalización (ISO) implementa el laboratorio de revisión de requerimientos para las

competencias de *testing de software*; ejercicio del cual se genera el documento ISO/IEC 17025: 2005, pautas a tener en cuenta que el *testing* debe demostrar técnicamente que el *software* es competente y capaz de generar los resultados esperados.

En Estados Unidos, en el Instituto Tecnológico de Massachusetts (MIT, por su sigla en inglés), el curso “Introducción al proceso personal de *software*” es una base para el programa de pregrado, y dedica un capítulo especial al proceso de pruebas de calidad de software (MIT, 2016), sin embargo, no se tiene un espacio dedicado al *testing* como tal.

En España, THOT, proyecto de innovación de la gestión documental aplicada a expedientes de contratación de servicios y obras de infraestructura de transporte (González, Mayo, Rodríguez y Cuaresma, 2014), propone la adaptación de un mecanismo ágil basado en la metodología Scrum para ejecutarse en las pruebas de aceptación del ciclo de desarrollo software.

En Beirut, Líbano, en la Universidad Americana de Beirut se realizó la indagación “Coverage-based software testing: beyond basic test requirements” (Masri y Zaraket, 2016), donde se examina por qué los requisitos deben ser tomados en cuenta para el proceso de pruebas; además, se menciona el gran valor de realizar pruebas tempranas y ahondar en el *testing* de requerimientos para disminuir el riesgo de fallos en el sistema. Se comparan a su vez dos generaciones de técnicas tomando como factor de medición la efectividad de las técnicas para revelar defectos.

En la Universidad Noruega de Ciencia y Tecnología, se trabajó en el proyecto “Challenges and strategies for motivating software testing personnel” (Deak, 2016); En el cual se realiza un estudio a través de entrevistas semiestructuradas y en profundidad, en 12 organizaciones con enfoque tecnológico en Noruega, delegaron a 36 personas como muestra para la investigación, con el objeto de analizar cómo el personal, las habilidades y la estimulación repercuten en la calidad de las aplicaciones que desarrollaban en

las organizaciones, proceso que trascendió en el tiempo de la investigación.

En Cuba, a través del Centro Nacional de Calidad de Software (Calisoft), se presenta un *software* para la gestión de calidad en un laboratorio de análisis y valoración de *software* en el 2013 con el cual se busca obtener la acreditación por parte del instituto de ciencia y tecnología. De ello nace el Laboratorio Industrial de Pruebas de Software en la Universidad de las Ciencias Informáticas de La Habana (Brito y Capote, 2013).

En Argentina, en Instituto de Tecnología Industrial-INTI en el año 2009 se crea el Laboratorio de Testing y Aseguramiento de Calidad de Software, “como parte del proceso de calidad de productos, realizando testing a todo tipo de productos”. Con esta implementación se crea un aporte sustancial en la comunidad de la localidad de Córdoba y alrededores en Argentina. (INTI, 2015). Es el caso más relevante de implementación de laboratorio de *testing* de *software* en el contexto académico.

Gasca, Camargo y Medina (2014) establecen una metodología para el desarrollo de aplicaciones móviles, y mencionan que una etapa denominada *pruebas de funcionamiento* es fundamental antes de la entrega de la aplicación y la prueba la clasifica en etapas de emulación y simulación, en dispositivos reales y análisis de los 6 modelos (uso, estructural, interacción, comportamiento, implementación y usabilidad).

Según Ardila, Pino, Pardo y Merchán (2014),

Una pequeña organización desarrolladora de software que no adopte prácticas de gestión cuantitativa tendrá dificultades para identificar y enfocarse en los factores que impactan en el rendimiento de los procesos y por consiguiente los productos que pudieran obtenerse a partir de ellos. (p.128).

En el contexto nacional, la Universidad Santiago de Cali cuenta con un laboratorio de computación móvil y banda ancha, cuyo objetivo principal es promover la investigación y desarrollo de la

computación móvil y banda ancha en Colombia (USC, 2018).

En su artículo Peña, Rodríguez, Pérez y Yobanis (2016) proponen un método para analizar la viabilidad técnica, comercial y social de los proyectos de software en entornos de incertidumbre.

En Ocaña a través de una investigación se toman dos niveles de madurez del CMMI y se analizan para establecer unos lineamientos de gobernanza de tecnología de información para la validación de proyectos en un laboratorio de auditoría en el año 2012 (Pérez, Velásquez y Pérez 2015)

En el ámbito nacional se evidencia que la investigación y la apropiación del *testing* de *software* es mínima, aunque la tendencia tecnológica y la industria TIC en Colombia se están potencializando.

En el contexto local, por su parte, en la Fundación Universitaria de Popayán se da inicio al laboratorio de informática con el propósito de mantenerse a la vanguardia tecnológica, aunque el laboratorio prestara servicios especializados a docentes y estudiantes en: desarrollo de software, simulación, redes, inteligencia artificial y robótica. Sin embargo, este no tiene un enfoque hacia el *testing* o las pruebas de calidad de *software*.

La ponencia presentada en el I Seminario de Calidad de Software, titulada “Mejora de procesos de software ágil con Agile Spi Process”, realizada por profesores de la Universidad del Cauca, con el fin de motivar a las mipymes (micro, medianas y pequeñas empresas) desarrolladoras de *software* iberoamericanas, en la exploración de un entorno de *software* competitivo con altos estándares de innovación, donde se tenga priorizada la calidad de *software* y los espacios para *testing* (Hurtado et al., s.f.).

Con un análisis del proceso de pruebas de calidad de *software*, Mera (2016) resalta que con el artículo genera resultados y conclusiones para que las compañías de desarrollo de productos de *software* mejoren el rendimiento, la eficacia y la optimización de los procesos de prueba de calidad del *software*.

Para Mera y Cano (2018), “es importante definir un marco de referencia o framework que permita realizar pruebas de calidad de software centrándose en términos de usabilidad” (p. 144).

Después de revisar el contexto académico, se evidencia que la existencia de laboratorios de *testing de software*, en relación con los procesos de formación académica, son mínimos. Aunque una gran parte de la comunidad universitaria reconoce que el *testing de software* y las actividades de pruebas de calidad son parte integral y necesaria en el desarrollo de *software*.

DISCUSIÓN DE LOS RESULTADOS

Al analizar las variables que inciden en el montaje para un laboratorio de *testing de software* para un contexto académico, se realizan unas preguntas a los directivos o encargados de los laboratorios de 10 instituciones académicas colombianas (se reservan nombres por petición de varias de ellas) que ofertan programas con enfoque de desarrollo de *software*:

Listado de preguntas

1. ¿Conocen el concepto de *testing* y control de calidad de *software*?
2. ¿Dentro de los programas de formación tendientes a desarrollo de *software*, emplean técnicas o modelos para *testing* y control de calidad de *software*?
3. ¿Cuentan con un espacio o infraestructura para el desarrollo de prácticas de *testing* o calidad de *software*?
4. ¿En su institución hay equipos especializados y destinados a recrear ambientes para el *testing de software*?
5. ¿Se documentan las prácticas de control de calidad o *testing*?
6. ¿En su institución, existe un laboratorio de *testing de software*?
7. Si contesta “no” en el punto 6. ¿Cuál de estos factores incide en que no exista el laboratorio?:
 - a. Falta de recursos económicos.

- b. Poco interés en la temática.
- c. Desconocimiento.
- d. Falta de espacio, infraestructura o equipos.

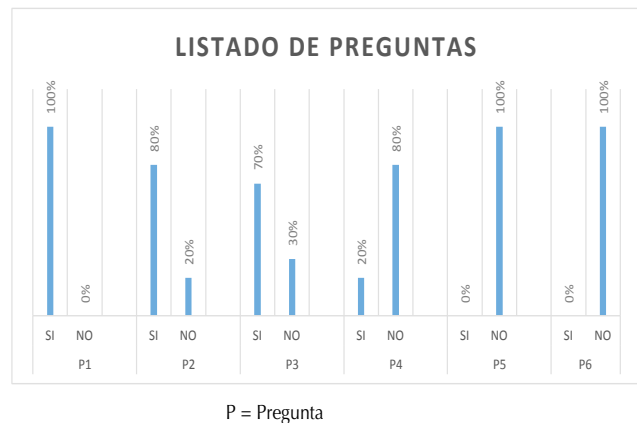


Figura 1. Listado de preguntas

Fuente: elaboración propia.

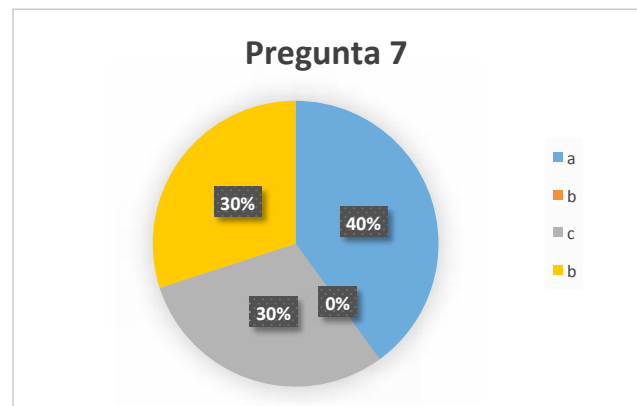


Figura 2. Resultados de pregunta 7

Fuente: elaboración propia.

A partir del cuestionario, se observa que hay interés por el *testing*; se considera importante en el desarrollo de *software*; se evidencia que un limitante es la falta de recursos económicos, equipos, espacio o infraestructura, y desconocimiento de la temática.

Más relevante que la información cuantitativa que se puede obtener de este cuestionario, es observar la relevancia cualitativa que tendría una

institución académica que en la región contara con un laboratorio de *testing* de *software*. La calidad académica de los profesionales egresados va a tener una relevancia significativa para la industria de *software* de la región.

CONCLUSIONES

El control de calidad de *software* y el *testing* cuentan con diferentes marcos de referencia, estándares y consorcios de investigación, además de congresos y comunidades académicas y no académicas que coinciden en brindar el valor adecuado que deben tener en el ciclo de vida de un producto *software*.

A través del diagnóstico tecnológico de la pertinencia de implementación de un laboratorio de *testing*, se evidencia que la fundamentación teórica, las necesidades de la industria y de la academia han estado desconectadas, al desarrollar *software* sin tener en cuenta la importancia y la relevancia de contar con condiciones de infraestructura, competencias y habilidades ligadas a generar un control de calidad de *software* adecuado y pertinente a las realidades políticas y sociales de la región. Por tanto, un laboratorio de *testing* de *software*, marca un hito de innovación en la Universidad Cooperativa de Colombia y en las instituciones de educación superior, puesto que genera un valor agregado a la calidad de formación de los programas con enfoque de sistemas o *software*, para la región y el país.

Se concluye que es vital la articulación entre la academia y los procesos reales de la creación industrial de *software* de talla mundial.

La normatividad, la revisión del contexto académico y la búsqueda literaria abren las puertas para adelantar investigaciones que permitan profundizar en la relación entre el entorno académico y el desarrollo de *software* en la industria; además, amplían el espectro de interpretación de modelos, métricas, factores de estudio, donde en un conjunto se apliquen a ambientes controlados y definidos, para llevar bitácoras, seguimiento e

informes, que permitan garantizar niveles de calidad óptimos.

Identificada la pertinencia de la implementación del laboratorio de *testing*, se plantea como trabajo futuro un análisis sistemático de información soportado en un trabajo de pregrado de ingeniería de sistemas, que determine las condiciones, requerimientos, recursos y competencias del talento humano, necesarios para la funcionalidad de un laboratorio con las características planteadas en la investigación.

REFERENCIAS

- Ardila Albarracín, C.A., Pino Correa, F.J., Pardo Calvache, C.J. y Merchán Paredes, L. (2014). MaTGeC: hacia un marco de trabajo para la gestión cuantitativa de procesos de desarrollo de software en pequeñas organizaciones. *Tecnura*, 18(42), 126-138. DOI: <https://doi.org/10.14483/udistrital.jour.tecnura.2014.4.a10>
- Brito, Y. y Capote, T. (2013). System Quality Management in Software Testing Laboratory that Chooses Accreditation. *Revista Antioqueña de las Ciencias Computacionales y la Ingeniería de Software*, 3(2) 13-18.
- Callejas-Cuervo, M., Alarcón-Aldana, A. C., & Álvarez-Carreño, A. M. (2017). Modelos de calidad del software, un estado del arte. *ENTRAMADO*, 13(1), 236-250. DOI: <https://doi.org/10.18041/entramado.2017v13n1.25125>
- Cooper, H. M. (1988). Organizing knowledge syntheses: A taxonomy of literature reviews. *Knowledge in Society*, 1(1), 104-126. DOI: <https://doi.org/10.1007/bf03177550>
- Constitución Política de Colombia (1991). 2a. ed. Bogotá: Legis.
- Deak, A., Stålhane, T. y Sindre, G. (2016). Challenges and strategies for motivating software testing personnel. *Information and Software Technology*, 73, 1-15. DOI: <http://dx.doi.org/10.1016/j.infsof.2016.01.002>
- Ewijk, A. V., Linker, B., Oosterwijk, M. V., & Visser, B. (2013). TPI next: business driven test process improvement. *Kleine Uil*.

- Furtado, A. P. C. C., Gomes, M. A. W., Andrade, E. C., & de Farias Junior, I. H. (2012). MPT.BR: A Brazilian Maturity Model for Testing. 2012 12th International Conference on Quality Software. DOI: <https://doi.org/10.1109/qsic.2012.53>
- García, C., Dávila, A. y Pessoa, M. (2014). Test Process Models: Systematic Literature Review. *Software Process Improvement and Capability Determination*, 477, 84–93. DOI: https://doi.org/10.1007/978-3-319-13036-1_8
- Gasca Mantilla, M.C., Camargo Ariza, L.L. y Medina Delgado, B. (2014). Metodología para el desarrollo de aplicaciones móviles. *Tecnura*, 18(40), 20-35. DOI: <https://doi.org/10.14483/udistrital.jour.tecnura.2014.2.a02>
- González, J.F.P., Mayo, F.J.D., Rodríguez, J.J.G. y Cuaresma, M.J.E. (2014). Pruebas de aceptación orientadas al usuario: contexto ágil para un proyecto de gestión documental. *Ibersid: Revista de Sistemas de Información y Documentación*, 8, 73-80.
- Hurtado, J.A., Pino, F.J., Vidal, J.C., Pardo, C., y Fernández, L.E. (s.f.). Agile SPI. *Software Applications*. DOI: <https://doi.org/10.4018/9781605660608.ch191>
- IEEE 1012 e ISO/IEC 29119: standards for software verification' (Healthcare Technologies, 2018), 'Engineering High Quality Medical Software: Regulations, standards, methodologies and tools for certification', Chap. 8, pp. 95-105, DOI: https://doi.org/10.1049/pbhe012e_ch8
- IET Digital Library, <https://digital-library.theiet.org/>;jsessionid=lxlkkjllhpffo.x-iet-live-01/content/books/10.1049/pbhe012e_ch8 Instituto de Tecnología Industrial (2015). "Laboratorio de Testing Córdoba", Córdoba, Argentina, Recuperado de: <http://www.inti.gob.ar>
- ISO/IEC 17025:2005 *General requirements for the competence of testing and calibration laboratories* [ISO].
- ISO/IEC 8402:1987 *Calidad. Vocabulario*. [ISO].
- International Software Testing Qualifications Board (ISTQB) (2016). *What is System Testing?*
- Lackner, H. y Schlingloff, B.-H. (2017). Advances in Testing Software Product Lines. *Advances in Computers*, 107, 157–217. DOI: <https://doi.org/10.1016/bs.adcom.2017.07.001>
- Masri, W. y Zaraket, F. A. (2016). Coverage-Based Software Testing. *Advances in Computers*, 79–142. DOI: <https://doi.org/10.1016/bs.adcom.2016.04.003>
- Massachusetts Institute of Technology (MIT) (2016). Program in Science, Technology, and Society. Recuperado de: <https://sts-program.mit.edu/>
- Mera Paz, J. (2016). Análisis del proceso de pruebas de calidad de software. *Ingeniería Solidaria*, 12(20), 163. DOI: <https://doi.org/10.16925/in.v12i20.1482>
- Mera Paz, J. y Cano Beltrán, J. (2018). Diagnóstico de pruebas de calidad en software para ambientes virtuales de aprendizaje sobre dispositivos móviles. *Memorias de Congresos UTP*, 1(1), 144-150. Recuperado de <http://revistas.utp.ac.pa/index.php/memoutp/article/view/1848>. DOI: <https://doi.org/10.2307/j.ctt2050wjh.9>
- Peña Abreu, M., Rodríguez Rodríguez, C.R., Pérez, P. y Yobanis, P. (2016). Computación con palabras para el análisis de factibilidad de proyectos de software. *Tecnura*, 20(50), 69-84.
- Pérez, T.V., Velásquez, A.M.P. y Pérez, Y.M.P. (2015). Un enfoque de buenas prácticas de gobierno corporativo de TI. *Tecnura*, 19(1), 159-169.
- Pressman, R.S. (2010). *Ingeniería del software, un enfoque práctico*. 7a. ed. Conenticut, USA: McGraw-Hill.
- Rowley, J. y Slack, F. (2004). Conducting a literature review. *Management Research News*, 27(6), 31–39. DOI:10.1108/01409170410784185
- Sanz, A., Saldaña, J., García, J. y Gaitero, D. (2008). Test PAI: A testing process area integrated with CMMI. In *Proceedings of the workshops of the Conference on Software Engineering and Databases, ICSE 2008 New Jersey, USA*.
- Universidad Cooperativa de Colombia (2013). *Plan estratégico nacional "Navegando juntos 2013-2022"*, Bogotá-Colombia. Recuperado de <http://www.ucc.edu.co/PlanEstrategico/Paginas/inicio.aspx>. DOI: <https://doi.org/10.16925/greylit.1754>
- Universidad Cooperativa de Colombia (2014). *Ambientes Prácticos de Aprendizaje con calidad práctica y académica*, Bogotá-Colombia.

Recuperado de: <https://www.ucc.edu.co/prensa/2014/Paginas/ambientes-practicos-de-aprendizaje-con-calidad-practica-y-academica.aspx>. DOI: <https://doi.org/10.22209/cice.n2a06>

Universidad Santiago de Cali (USC) (2018). *Descripción de los laboratorios*. Recuperado de <http://www.usc.edu.co/index.php/descripcion-de-los-laboratorios>

Van Veenendaal, E. y Wells, B. (2012). Test maturity model integration (TMMi). TMMI Foundation (www.tmmifoundation.org), *Uitgeverij Tutein Nolthenius*.

Vom Brocke, J., Simons, A., Riemer, K., Niehaves, B., Plattfaut, R. y Cleven, A. (2015). Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research. *Communications of the Association for Information Systems*, 37. DOI: <https://doi.org/10.17705/1cais.03709>

Wong, W.E. (2016). Special Section on Software Quality Assurance: Research and Practice. *IEEE Transactions on Reliability*, 65(1), 3-7.

