

# Estrategia didáctica para la formación de pruebas de software en el profesional informático.

## Didactic strategy for the training of software testing in the IT professional.

Darío Enrique Soto-Durán<sup>1</sup>, Aixa Eileen Villamizar-Jaimes<sup>2</sup>, LF Bohórquez-Chacón<sup>3</sup>,  
Fabio Alberto Vargas-Agudelo<sup>4</sup>, Maria Valeria Poliche<sup>5</sup>, MA Amaya-Mancilla<sup>6</sup>  
<sup>1,2,4</sup> *Tecnológico de Antioquia, Medellín - Colombia*, <sup>3,6</sup> *Universidad de Santander, Cúcuta - Colombia*,  
<sup>5</sup> *Universidad Nacional de Catamarca, Catamarca - Argentina*  
ORCID: <sup>1</sup>[0000-0002-6557-844X](https://orcid.org/0000-0002-6557-844X), <sup>2</sup>[0000-0002-4070-1763](https://orcid.org/0000-0002-4070-1763), <sup>3</sup>[0000-0002-8500-3475](https://orcid.org/0000-0002-8500-3475), <sup>4</sup>[0000-0002-6921-4918](https://orcid.org/0000-0002-6921-4918),  
<sup>5</sup>[0000-0001-9074-035X](https://orcid.org/0000-0001-9074-035X), <sup>6</sup>[0000-0002-8157-0288](https://orcid.org/0000-0002-8157-0288)

Recibido: 06 de julio de 2022.

Aceptado: 23 de agosto de 2022.

Publicado: 01 de septiembre de 2022.

**Resumen-** La formación en pruebas de software es relevante en los diferentes niveles de formación del Ingeniero, dado que esta actividad es intensiva en conocimiento y pertinente para la calidad de los productos de Software. Siendo necesario una definición de competencias que logre establecer salidas ocupacionales diferenciadas en campo laboral. En la industria del software es necesario reconocer las competencias para definir perfiles de formación de los programas académicos asociados a la cadena de valor de las pruebas de software a través de estrategias de formación acordes a las demandas del sector productivo y los retos de la educación superior. Por ello, este estudio presenta resultados de un proceso, que valida con la academia y la industria los perfiles ocupacionales y competencias del proceso de pruebas de software para establecer un proceso de enseñanza aprendizaje centrado en el estudiante a través de un entorno educativo que instancia el modelo de aprendizaje experiencial y la cultura maker.

**Palabras clave:** pruebas de software; educación de Ingeniería; competencias; aprendizaje experiencial; estrategia didáctica.

**Abstract—** Training in software testing is relevant at different levels of engineering education, since this activity is knowledge-intensive and relevant to the quality of software products. A definition of competences is necessary to establish differentiated occupational outlets in the labor field. In the software industry it is necessary to recognize the competences to define training profiles of the academic programs associated to the software testing value chain through training strategies according to the demands of the productive sector and the challenges of higher education. Therefore, this study presents the results of a process that validates with the academy and the industry the occupational profiles and competencies of the software testing process to establish a student-centered teaching and learning process through an educational environment that calls for the experiential learning model and the maker culture.

**Keywords:** software testing; engineering education; competences; experiential learning; didactic strategy.

\*Autor para correspondencia.

Correo electrónico: [aixa.villamizar@tdea.edu.co](mailto:aixa.villamizar@tdea.edu.co) (Aixa Eileen Villamizar Jaimes).

La revisión por pares es responsabilidad de la Universidad de Santander.

Este es un artículo bajo la licencia CC BY (<https://creativecommons.org/licenses/by/4.0/>).

Como citar este artículo: D. E. Soto-Durán, A. E. Villamizar-Jaimes, L. F. Bohórquez-Chacón, F. A. Vargas-Agudelo, M. V. Poliche y M. A. Amaya-Mancilla, "Estrategia didáctica para la formación de pruebas de software en el profesional informático", *Aibi revista de investigación, administración e ingeniería*, vol. 10, no. 3, pp. 1-12, 2022. doi: [10.15649/2346030X.3017](https://doi.org/10.15649/2346030X.3017)

## I. INTRODUCCIÓN

El ingeniero en software tiene como propósito el estudio y aplicación de los métodos y procesos asociados al ciclo de vida del software con el objeto de asegurar la calidad del producto.

La calidad de software se entiende como el cumplimiento de los requisitos y/o necesidades para el cual se construye el producto; asimismo, que sus atributos lo hacen útil para las personas. Siendo, el propósito de alto nivel de la ingeniería de software lograr productos de software con calidad. En consecuencia, el aseguramiento de la calidad del software y la política de pruebas se convierte en un elemento relevante de la calidad. El proceso formativo de los profesionales asociados al desarrollo de software debe incorporar competencias en pruebas que permitan un personal formado y cualificado en las políticas y herramientas.

En el sector del software a nivel nacional e internacional en ámbito de las pruebas de software se promueven diferentes cursos y certificaciones que buscan promover la formación y cualificación de los profesionales en esta área.

En el mundo laboral las pruebas de software demandan un gran número de personal, en las diferentes funciones ocupacionales del proceso de construcción. Por ello la industria del software desarrolló el estándar ISO/IEC 29119, con el objetivo unificar diferentes enfoques que ya existían de la disciplina. Sin embargo, a nivel académico no existe claridad en las competencias que se deben incorporar en el proceso de enseñanza en el contexto universitario.

## II. MARCO TEÓRICO

A continuación, se presentan los conceptos y experiencias relacionadas con el objeto de estudio. Inicialmente se describen los conceptos asociados a las pruebas de software y al desarrollo de competencias. Por último, se presenta un análisis de las experiencias asociadas a la enseñanza de pruebas de software en el contexto académico.

### 1. Pruebas de Software

Según la IEEE (Institute of Electrical and Electronics Engineers), las pruebas de software, es: “la actividad en que un sistema o un componente se ejecuta bajo condiciones controladas, los resultados son registrados y la evaluación es realizada sobre algún aspecto del sistema o componente” [1]. Dado las innovaciones y diferentes enfoques disciplinares que soportan las pruebas de software, surge la IEEE 29199 que integra los estándares: IEEE 829, IEEE 1008 y BS 7925, como referente que describe el lenguaje, procesos, entregables, técnicas y un modelo para evaluar el proceso de pruebas independiente del ciclo de vida de desarrollo del software.

Acorde con la IEEE 29119, las pruebas integran los componentes de verificación y validación. Lo cuales asumen dos características de la prueba: el nivel y tipo [2]. El nivel se refiere al alcance del producto sometido a evaluación (unitarias, componente, sistema y aceptación) y tipo tiene como base el aspecto que se desea evaluar (seguridad, funcionalidad entre otros).

### 2. Competencias

El concepto de competencia de manera clásica se reconoce como la integración entre el saber, saber hacer y saber ser [3], [4]. Sin embargo, según el aspecto que se desee abordar. Para nuestro caso se aborda el concepto desde lo ocupacional y/o profesional y los resultados de aprendizaje. A nivel ocupacional el término de competencia se interpreta como la capacidad del profesional para realizar una función en el contexto empresarial. Los resultados de aprendizaje (RA) se convierten en factor esencial de los programas académicos de educación superior, denotando la meta que este debe alcanzar en función de un proceso integrado entre enseñanza aprendizaje y evaluación, con el estudiante como el eje del proceso formativo [5].

En este contexto el nivel de desarrollo de la competencia solo se alcanza al máximo cuando se tiene mayor grado de experticia. Los autores como [6] y [7] coinciden que los atributos de una competencia se derivan de la articulación entre conocimientos, actitudes, aptitudes y destrezas para desarrollar una tarea o función.

Según el proyecto [8], las competencias en el marco educativo se conciben como “los conocimientos, habilidades, actitudes y responsabilidades, que describen los resultados del aprendizaje de un programa educativo o lo que los alumnos son capaces de demostrar al final del proceso educativo”. Las competencias se clasifican en específicas y genéricas. Las especificadas son asociadas al objeto de estudio y genéricas a las que se derivan de las condiciones requeridas en el ámbito profesional o también propias de la formación integral.

### 3. Experiencias previas

De acuerdo con [9] en los diferentes estudios sobre la definición de competencias en la formación del profesional informático se puede evidenciar lo siguiente:

- Existe un bajo porcentaje de contenidos asociados a los procesos de pruebas en los pensum académicos.
- Las empresas afirman que los profesionales de egresados no tienen competencias en el dominio específicos de pruebas de software.
- Existe mayor interés en las técnicas y procesos de construcción de software que en el aseguramiento de calidad.
- La ausencia de la formación específica en pruebas de software ha generado la demanda de certificaciones como ISTQB (International Software Testing Qualifications Board) y TMMI (Testing Maturity Model integrated).
- Las estrategias didácticas no son pertinentes con los escenarios requeridos en el contexto real.
- Se requiere una estrategia de formación progresiva y continua.

En el contexto de la industria 4.0, [10] identifican competencias requeridas para los profesionales en este campo, tales como: la interdisciplinariedad del conocimiento, la resolución de problemas, la interculturalidad, y el aprendizaje continuo. Igualmente, [11] desde una perspectiva del sistema de educación superior, la formación en ingeniería requiere fortalecer factores como: la flexibilidad, la interdisciplinariedad, las alianzas universidad empresa y sistemas de aprendizaje abierto, soportados en plataformas de aprendizaje y cursos de acceso gratuito.

Por ello, se requiere cambiar el centro del proceso de enseñanza del profesor al estudiante, cambiando prácticas y enfoques donde el estudiante es el receptor de información [12] dado que la sola transmisión de información ya no caracteriza un proceso eficiente de enseñanza y aprendizaje en la sociedad interconectada y globalizada [13].

Siendo relevante las metodologías activas para lograr que el estudiante sea el eje de formación y protagonista en la construcción del proceso de aprendizaje, de manera flexible, interconectada e híbrida [13]. Entre las estrategias metodológicas más usadas se referencian: el aprendizaje basado en problemas, el aprendizaje soportado en proyectos, el aula invertida, el aprender haciendo y aprendizaje basado en equipos.

Asimismo, las metodologías activas y las tecnologías digitales generan valor agregado en el proceso de formación, incentivando el pensamiento crítico, la innovación, la solución de problemas entre otros.

En este marco de metodologías activas se identifica el modelo de aprendizaje experiencial de Kolb [14], el cual se basa en la apropiación de conocimiento a partir de la transformación de la experiencia, siendo importante, establecer un escenario de aprendizaje para que el estudiante descubra el conocimiento a partir de su propia vivencia. En este se consideran tres fuentes de aprendizaje; aprendizaje del contenido, aprendizaje de la experiencia y aprendizaje de la reflexión [15].

El modelo de Kolb en el contexto del software [16], está sustentado en las siguientes fases: (i) La Experiencia Concreta; momento donde se abordan los conceptos teóricos, (ii) La Observación Reflexiva; momento para la reflexión consciente sobre el concepto o la teoría, (iii) La Conceptualización Abstracta; momento para establecer argumentos y principios para la aplicación del concepto, y finalmente (iv) La validación Activa; momento para la comprobación de las teorías, prácticas y procesos, como parte de aplicación práctica.

Lo anterior conlleva a considerar también la necesidad de abordar las bases teóricas del aprendizaje individual y grupal en el marco de las organizaciones y desde las perspectiva de la gestión de Conocimiento; como ese proceso natural de introspección, apropiación y reflexión que se genera a partir de la experiencia vivida del individuo, pero que en el marco de las organizaciones se vuelve consciente y se valida en grupo a partir del dialogo y el aprendizaje de rutinas compartidas y aceptadas por el equipo de trabajo. De acuerdo con los autores [17], [18], [19], con relación a los subprocesos del aprendizaje individual, se podría interpretar que, en el entorno de aprendizaje en los escenarios organizacionales de desarrollo de software, se dan los momentos representados en la figura 1. Subprocesos del Aprendizaje Individual en el entorno de aprendizaje de desarrollo.

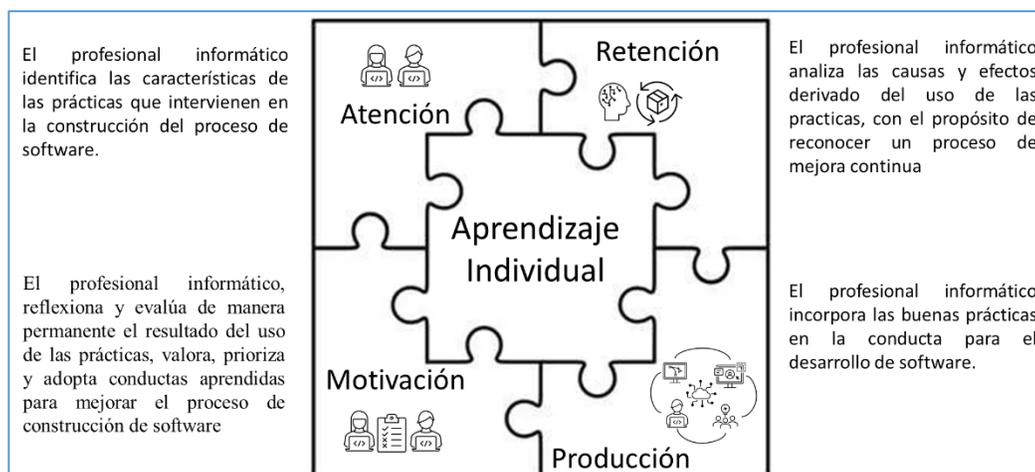


Figura 1: Subprocesos del Aprendizaje Individual en el entorno de aprendizaje de desarrollo. Fuente: Elaboración propia.

### III. METODOLOGÍA O PROCEDIMIENTOS

El enfoque de la investigación es mixto de alcance descriptivo de acuerdo con la naturaleza de los datos recolectados y las técnicas utilizadas tanto para la recolección como el análisis de datos; mediante las cuales se establece una estrategia didáctica para la enseñanza aprendizaje de las pruebas de software. La ruta metodológica se presenta en la figura 1.

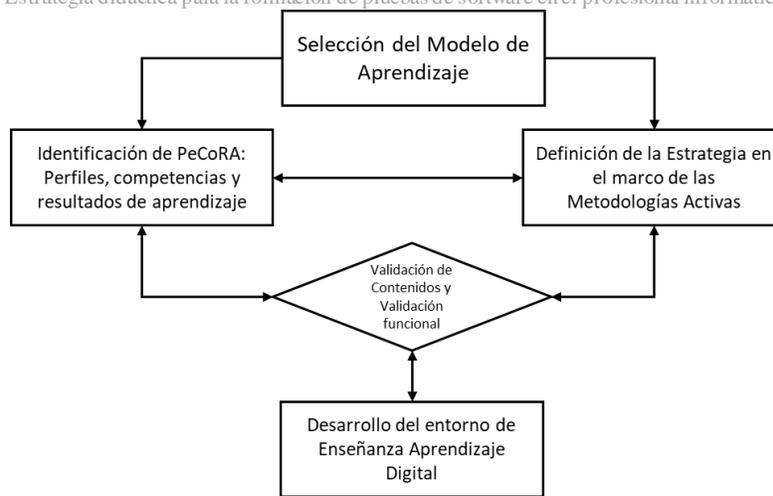


Figura 2: Ruta metodológica.  
Fuente: Elaboración propia.

La población está conformada por docentes de instituciones de educación superior, desarrolladores y representantes de empresas de dos ciudades colombianas y una Argentina, con quienes se validaron las competencias identificadas en la revisión de literatura e identificaron las habilidades requeridas por un profesional en dominio de Pruebas de Software en el ámbito empresarial. Así mismo por Jóvenes entre los 16 y 30 años con quienes finalmente se ejecutaron pruebas de aceptación del entorno de enseñanza aprendizaje digital planteado para el modelo. La muestra finalmente estuvo conformada por 16 representantes de empresas, 11 docentes, 12 Estudiantes de Ingeniería y de tecnología en Software.

Para la identificación previa de competencias se plantearon las siguientes preguntas: Cuáles eran los referentes que describen la disciplina pruebas de software, Cuáles eran las estrategias implementadas en el contexto educativo y empresarial para desarrollar competencias requeridas en el área de pruebas de software y Cuáles era las competencias en pruebas de software necesarias para un profesional en el área de informática por niveles de formación: técnico, tecnólogo e ingeniero.

Se recurre a la Revisión Sistemática de la Literatura, la encuesta, el grupo focal, historias de usuario, análisis descriptivo, listas de cotejo, análisis de contenido, validación por juicio de expertos, técnicas ágiles de desarrollo software y pruebas de software.

Con el fin de conocer el estado actual de las funciones y roles que desempeñan los profesionales informáticos en las organizaciones que incorporan pruebas de software, se diseña una encuesta estructurada en 5 preguntas basado en los siguientes conjuntos:

- Conjunto 1: Tener una percepción sobre los roles más comunes en la industria y el grado de aceptación de las competencias definidas en la investigación por cada nivel de formación. (P1 y P2).
- Conjunto 2: Conocer los tipos de pruebas de software que más se usan a nivel productivo según el nivel y aspecto. (P3) e identificar las herramientas que se usan en la industria para automatizar y gestionar el proceso de pruebas de software. (P4 y P5).

La encuesta fue dirigida a empresas de software instaladas en las ciudades de Medellín (Colombia) y Catamarca (Argentina). La Tabla 1. Muestra la Ficha técnica de la encuesta.

Tabla 1: Ficha técnica de la encuesta.

<b>Población Objetivo</b>	Profesionales vinculados a las empresas del sector del software de la ciudad de Medellín (Colombia) y de empresas en operación en Argentina, que realizan actividades de pruebas de software y que aceptaron participar en el estudio.
<b>Método de Recolección de Datos</b>	Encuestas digitales
<b>Tamaño de la muestra:</b>	16 profesionales
<b>Selección de la muestra</b>	Intencional no probabilística por conveniencia
<b>Periodo de aplicación de encuestas</b>	16/06/2021 – 20/06/2021

Fuente: Elaboración propia.

## IV. RESULTADOS

### 1. Fase 1. Selección del modelo de aprendizaje (Aprendizaje Experiencial e Individual)

Partiendo del hecho que las pruebas de software son una actividad soportada en conocimiento que articula la apropiación de los conceptos y teorías a partir de un proceso práctico según [20], se instancia un modelo de aprendizaje basado en [14], [17], [18]. Se plantea un modelo, donde los momentos de aprendizaje experiencial e individual, basados en los marcos estudiados, se desarrollen en cuatro momentos denominados: Aprende, Practica, Aplica y Evalúa. La tabla 2 intenta alinear los momentos del entorno de aprendizaje del modelo propuesto con los encontrados en los marcos de referencia.

Tabla 2: Modelo de aprendizaje vs Entorno de aprendizaje.

Aprendizaje Experiencial	Aprendizaje Individual	Momentos del entorno de Aprendizaje	Descripción
Experiencia Concreta	Atención	Aprende	Conceptualización (siglas, abreviaturas, contextualización, contenido de aprendizaje, síntesis, temas relacionados)
Observación Reflexiva	Retención	Practica	Proceso de control de aprendizaje reflexión consciente de los conceptos, logrando la retención requerida (test de Conocimiento)
Conceptualización Abstracta	Producción	Aplica	Práctica del proceso de la prueba de Software asistida, autónoma o en equipos (estudios de caso, laboratorio de prueba, documentación), establecimiento de argumentos y principios para la aplicación de los conceptos, incorporar buenas prácticas para el desarrollo de software
Validación Activa	Motivación	Evaluación	Evaluación y retroalimentación del aprendizaje individual, contrastación

Fuente: Elaboración propia.

## 2. Fase 2. Identificación de PeCoRA: perfiles, competencias y resultados de aprendizaje requeridos en el proceso de pruebas de software.

La revisión de literatura en relación a las preguntas planteadas sobre las pruebas de software, factores o referentes que impactan el proceso de pruebas de software y estrategias implementadas en el contexto educativo y empresarial para desarrollar competencias requeridas, generan los siguientes resultados.

### a. Referentes que describen la disciplina pruebas de software.

La tabla 3. Muestra los referentes encontrados y asumidos como fuentes que describen la disciplina de pruebas de Software. Se intenta mostrar en la síntesis, el referente, los factores clave de éxito en la formación para el profesional informático, el perfil y las competencias o resultados de aprendizaje interpretados.

Tabla 3: Referentes, Factores Clave de formación de perfiles y competencias.

Referente	Factores Clave	Perfiles del Profesional Informático/Competencias/Resultados de Aprendizaje
Libro Azul: [21]	Validez en términos de funcionalidad, consistencia de la información, desempeño de infraestructura	<i>Ingeniero para la Gestión de Proyectos.</i> Pruebas de software como competencia específica en el diseño y desarrollo de proyectos. Se realiza la prueba para comprobar la validez del proyecto en términos de funcionalidades consistencia de información, desempeño de la infraestructura que soporta la aplicación entre otros aspectos.
IEEE: De acuerdo con (IEEE Computer Society) [22]	Calidad del proceso y del producto de Software, entorno de la prueba	<i>Ingeniero de Software.</i> Verificación y validación de software como competencias fundamentales para desarrollar un producto de alta calidad. Incluyen todas aquellas que estén asociadas con garantizar un proceso de calidad y a su vez un producto de calidad, haciendo énfasis tanto en el aseguramiento como en el control de la calidad del software
SWBOK: Según la Guía SWEBOK. [23]	Transversalidad de la prueba durante el ciclo de vida del desarrollo y mantenimiento del software	<i>Profesional Informático.</i> Las pruebas de software en los diferentes momentos del ciclo de vida del desarrollo y mantenimiento del software y no posterior al proceso de codificación. Aplicación de técnicas de prueba asociada a las competencias, capacidades y experiencias del profesional informático, al tipo de producto de software y la naturaleza de la aplicación
ACM:(ACM/IEEE9 Guía curricular para programas de pregrado en Ingeniería de Software [24]	Verificación y validación de software en función de la satisfacción de requisitos y cumplimiento de las expectativas del Cliente.	<i>Ingeniero de Software.</i> Competencias orientadas a la aplicación de los diferentes tipos y niveles de prueba (unidad, integración, sistemas y aceptación), al mismo tiempo al uso de diferentes técnicas y herramientas para el desarrollo de pruebas apropiadas con el fin de realizar, analizar y planificar casos de prueba en cada momento del ciclo de vida del software.

Fuente: Elaboración propia.

### b. Estrategias implementadas en el contexto educativo y empresarial para desarrollar competencias requeridas en el área de pruebas de software

De acuerdo con [25], [26] sugiere como estrategia didáctica, la incorporación de un laboratorio de proyectos, en el cual se incorpora una simulación del proceso de desarrollo en un contexto empresarial. Esta experimentación se enfocada en el análisis del proceso de calidad del software en ambientes similares a una fábrica de software. Presentar a los estudiantes durante el proceso de formación, las mejores prácticas estándar de la industria del Software, permite aumentar la motivación y el interés de los estudiantes en esta área, y mejorar su éxito académico y su mercado laboral.

[27] propone una estrategia didáctica basada en la experimentación de escritura de código para el aprendizaje de pruebas unitarias, identificación de defectos, y asociar dichos defectos a un caso de prueba específico. Se implementan herramientas de generación de código aleatorio, en el cual se agregan defectos y son compartidos con los estudiantes mediante actividades enfocadas en la detección de fallos con el fin de incentivar en el estudiante la competencia necesaria para detectar errores durante la prueba y documentarla. Por otra parte, [28] plantean una estrategia enfocada en la automatización de pruebas de software para disminuir la complejidad de las pruebas tanto en el ámbito universitario y en el laboral.

Esta propuesta está conformada por diferentes niveles, en donde se definen las pruebas y técnicas a utilizar y se definen los objetivos respectivos. Esta estrategia es validada en proyectos reales con el fin de alcanzar los resultados esperados. La tabla 4 muestra los niveles y las técnicas propuestas a aplicar.

Tabla 4: Técnicas aplicadas por niveles.

Nivel	Actores	Técnicas usadas	Objetivos
Nivel 1: Prueba Inicial	Equipo de Desarrollo	Caja Blanca	Validar las soluciones a nivel de pieza de código o módulo.
Nivel 2: Prueba de Sistema	Equipo de Pruebas	Caja Negra	Contrastar la funcionalidad a partir de los requisitos y resultados esperados.
Nivel 3: Prueba de Aceptación	Equipo de Pruebas Clientes	Caja Negra	Comprobar el cumplimiento de los criterios funcionales a partir de las reglas de negocio definidas los procesos de negocio previsto por el cliente y definidas en los requerimientos para su construcción.
Nivel 4: Prueba de Explotación	Equipo de Pruebas	Caja Negra	Comprobar que el desempeño de aplicación se desarrolla acorde con las condiciones esperadas en el ambiente de producción.

Fuente: Elaboración propia.

**c. Competencias en pruebas de software necesarias para un profesional en el área de informática por niveles de formación: técnico, tecnólogo e ingeniero.**

A partir de la información recopilada y con base a los referentes de las pruebas de software anteriormente mencionados, se establecen las competencias genéricas de acuerdo con el nivel de formación, las cuales se diferencian en la Tabla 2.

Tabla 5: Competencias en pruebas de software según el nivel de formación.

Competencias en pruebas de software	Niveles de formación		
	Técnico	Tecnólogo	Ingeniero
Aplicar técnicas para ejecutar conjuntos de pruebas con alta probabilidad de encontrar errores.	X		
Ejecutar pruebas usando herramientas para diferentes fases del ciclo de vida del testing.	X		
Recopilar y generar informes asociados a la ejecución de los casos de prueba	X		
Diseñar y preparar el entorno y la implementar las condiciones requeridas en los casos de prueba.		X	
Evaluar los resultados de la ejecución de las pruebas para identificar el reproceso apropiado.		X	
Planificar los recursos disponibles para lograr los objetivos de las pruebas		X	
Evaluar la madurez de las pruebas en una organización y mejorarla de forma paulatina.			X
Analizar los resultados de las pruebas para identificar oportunidades apropiadas de mejoras de procesos.			X
Coordinar y dirigir las actividades de las pruebas en una organización en todos los procesos fundamentales (Ciclo de vida de las pruebas de software).			X

Fuente: Elaboración propia.

**d. Estado actual de las funciones y roles que desempeñan los profesionales informáticos en las organizaciones: Tabulación y análisis de resultados de las encuestas**

Al realizar la tabulación de la encuesta se obtiene los siguientes resultados:

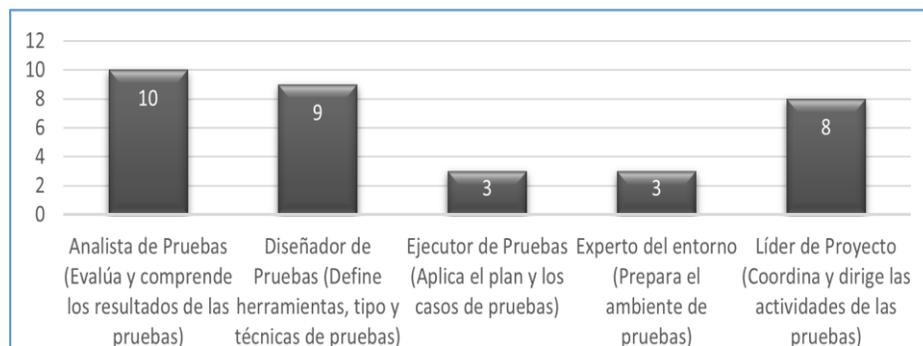


Figura 3: Pregunta #1 Roles o funciones en el ámbito de pruebas.

Fuente: Elaboración propia.

De acuerdo con los resultados, se puede apreciar que los roles más ejercidos en el ámbito de pruebas de software son los de analista, diseñador y líder de proyectos de pruebas obteniendo una calificación más alta con respecto a las otras.

A la pregunta, según su criterio, ¿Qué tan necesarias son las competencias que se definen a continuación para cada nivel de formación? Teniendo en cuenta la escala de valores, donde 1 es bajo y 5 es el valor más alto. Las respuestas ante el perfil de un Técnico en sistemas se pueden observar en la Tabla 5.

Tabla 6: Competencias a nivel Técnico profesional.

Competencias/Nivel	1	2	3	4	5
Aplicar técnicas para ejecutar conjuntos de pruebas con alta probabilidad de encontrar errores.	0	0	4	5	8
Ejecutar pruebas usando herramientas para diferentes fases del ciclo de vida del testing.	0	0	4	9	2
Recopilar y generar informes asociados con la ejecución de casos de prueba	0	0	4	9	2

Fuente: Elaboración propia.

Los resultados mostrados se encontraron que las competencias definidas para un técnico profesional son consideradas en un nivel sobresaliente por parte de los encuestados, obteniendo una escala de valor superior a 3 en la mayoría de las respuestas. Los resultados para el perfil de un Tecnólogo en sistemas se pueden observar en la Tabla 6.

Tabla 7: Competencias a nivel tecnólogo.

Competencias/Nivel	1	2	3	4	5
Diseñar y preparar el entorno y los casos de prueba.	0	0	4	5	6
Evaluar los resultados de la ejecución de las pruebas para identificar los reprocesos.	0	0	3	5	7
Planificar los recursos disponibles para lograr los objetivos de las pruebas	0	1	4	3	7

Fuente: Elaboración propia.

La apreciación de las competencias definidas para los tecnólogos muestra una tendencia similar a las del técnico, lo cual se interpreta como competencias aceptadas o con un buen concepto por parte de los empresarios.

Tabla 8: Competencias a nivel de formación de Ingeniería.

Competencias/Nivel	1	2	3	4	5
Evaluar la madurez de las pruebas en una organización y mejorarla de forma paulatina.	0	0	2	1	12
Analizar los resultados de las pruebas para identificar oportunidades apropiadas de mejoras de procesos.	0	1	1	2	11
Coordinar y dirigir las actividades de las pruebas en una organización en todos los procesos fundamentales (Planificación y control, Análisis y diseño, implementación y ejecución, Evaluación de criterios de salida y reporte, cierre de las pruebas).	0	2	3	8	2

Fuente: Elaboración propia.

Se puede apreciar en la Tabla 7 que las competencias que se definieron para un ingeniero igualmente tienen niveles de aceptación altos.

Tabla 9: Relación de resultados con respecto al nivel de pruebas según el aspecto.

Pruebas	Cantidad	Porcentaje
Confiabilidad	2	6,5%
Funcionales	5	16%
Rendimiento	5	16%
Seguridad	10	32%
Usabilidad	9	29%

Fuente: Elaboración propia.

Según las respuestas de los encuestados las pruebas más utilizadas teniendo en cuenta el aspecto son las pruebas de seguridad y usabilidad con un 32% y 29% respectivamente.

Tabla 10: Relación de resultados con respecto al nivel de pruebas según el tipo.

Pruebas	Cantidad	Porcentaje
Aceptación	6	15,8
Componentes	7	18,4
Integración	11	28,9
Regresión	4	10,5
Sistema	6	15,8
Unitarias	4	10,5

Fuente: Elaboración propia.

Teniendo en cuenta el nivel, las pruebas más utilizadas, según los encuestados, son las pruebas de integración con un porcentaje del 28.9% y las de componentes con un porcentaje del 18.4%.

Con relación a las herramientas que se usan en la industria para automatizar y gestionar el proceso de pruebas de software, las figuras 4 y 5 muestran los resultados obtenidos de las siguientes preguntas respectivamente: ¿Realiza automatización de pruebas a nivel de Gestión del proceso? Si la respuesta anterior es si, seleccione las Herramientas que usa: (Fitnesse, Mockito, QAManager, TestLink).

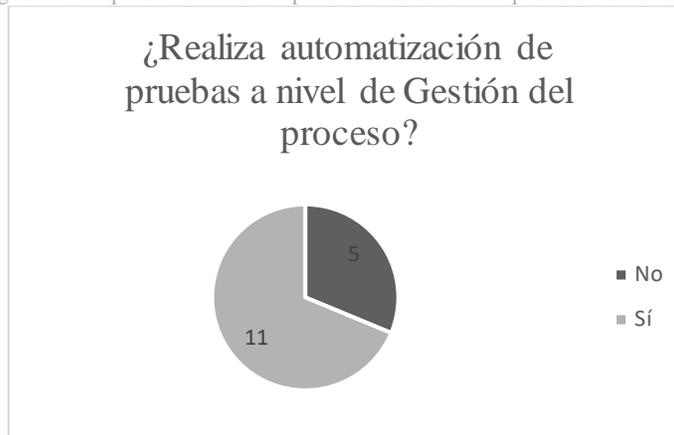


Figura 4: Pregunta 4.  
Fuente: Elaboración propia.

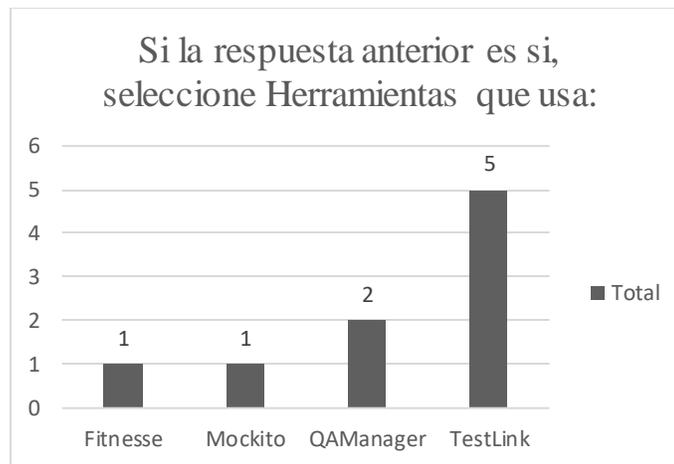


Figura 5: Pregunta 5.  
Fuente: Elaboración propia.

Se puede percibir, según las respuestas de los encuestados, el uso de herramientas para gestionar el proceso de Pruebas de Software a nivel del proceso no es muy común, pues la mayoría de los encuestados manifiestan no usar estas herramientas, y los que usan este tipo de pruebas utilizan TestLink y QAManager para gestionar su proceso de pruebas de software.

#### e. Resultados de los grupos focales

Posterior a la encuesta, se realiza un cotejo de las competencias definidas a partir de los resultados de la encuesta con las recomendaciones por niveles de la ISO 29119, incorporando 3 competencias más, las cuales fueron confirmadas con empresarios y docentes en grupos focales, diferentes a los participantes involucrados en la encuesta previa. El resultado permite redefinir 12 competencias de acuerdo con los factores claves establecidos en la norma, e identificados en los referentes analizados. Estos factores clave a incorporar en el proceso de la prueba y por tanto en la formación de competencias son: casos de prueba, el contexto de la organización, requisitos del producto, entorno de la prueba y requisitos de calidad del software, en correspondencia a las recomendaciones de la ISO/IEC/IEEE 29119-1, 29119-2, 29119-3 y 29119-4. Así mismo se determinó el conjunto de objetivos de aprendizaje a alcanzar por cada competencia con base en la complejidad de esta. Las tablas 10 y 11 muestran respectivamente las competencias redefinidas 10 de los 27 Objetivos de aprendizaje planteados a partir de los grupos focales y las competencias asociadas. La Lista continua a la tabla 10 describe los Objetivos de Aprendizaje planteados.

Tabla 11: Competencias definidas.

Competencias	Lineamiento ISO
C1. Definir los artefactos de mayor relevancia a usar en las pruebas de software de acuerdo con requisitos del producto y articulados al contexto de la organización.	ISO/IEC/IEEE 29119-1:2013
C2. Planificar los recursos disponibles para lograr los objetivos de las pruebas.	ISO/IEC/IEEE 29119-2:2013
C3. Diseñar y preparar el entorno y los casos de prueba basado en los requisitos del producto de software.	ISO/IEC/IEEE 29119-4:2015
C4. Definir las plantillas de entrada y salida del proceso de pruebas de software incorporando los diferentes parámetros de ejecución del producto de software.	ISO/IEC/IEEE 29119-3:2013
C5. Liderar las actividades de las pruebas en una organización en todos los procesos fundamentales (Planificación y control, Análisis y diseño, implementación y ejecución, Evaluación de criterios de salida y reporte, cierre de las pruebas)	ISO/IEC/IEEE 29119-2:2013
C6. Diseñar el plan de pruebas de software y gestión que atienda a los objetivos de calidad de desarrollo en una organización	ISO/IEC/IEEE 29119-2:2013

Competencias	Lineamiento ISO
C7. Definir las técnicas para la elaboración de los casos pruebas y los reportes del proceso basadas en las especificaciones del plan de prueba y en la experiencia	ISO/IEC/IEEE 29119-4:2015
C8. Evaluar los resultados de la ejecución de las pruebas para identificar el reproceso apropiado	ISO/IEC/IEEE 29119-3:2013
C9. Analizar los resultados de las pruebas para identificar oportunidades apropiadas de mejora de procesos	ISO/IEC/IEEE 29119-2:2013
C10. Evaluar la madurez de las pruebas en una organización y mejorarla de forma paulatina	ISO/IEC/IEEE 29119-3:2013
C11. Analizar el contenido de las normas ISO 29119 para su implementación en las organizaciones.	ISO/IEC/IEEE 29119-1:2013
C12. Implementar mecanismo de automatización e integración de software	ISO/IEC/IEEE 29119-4:2015

Fuente: Elaboración propia.

### Objetivos de Aprendizaje planteado

- OBJ1: Comprender el concepto de artefacto, prueba de software, error, fallo, defecto y riesgo, contando con los criterios para identificarlos y diferenciarlos en un caso práctico
- OBJ2: Comprende el propósito de las pruebas
- OBJ3: Comprender los procesos de V&V validación y verificación fundamentales en la calidad del producto y los utiliza en el análisis de un proceso de prueba de software.
- OBJ4: Conocer la importancia de la implementación del modelo en V en PS como una técnica reconociendo sus fases, ventajas y desventajas; con los criterios para determinar las fases que se deben llevar a cabo en el plan de pruebas para ejecutar un modelo en V.
- OBJ5: Conocer el modelo W, su importancia en las pruebas de software, la diferencia que existe entre el modelo V y el modelo W, con los criterios para determinar las fases que se deben llevar a cabo en el plan de pruebas para ejecutar un modelo en W.
- OBJ6: Comprende los beneficios de la implementación de pruebas estáticas y sus principales funciones en el desarrollo de pruebas de software, con los criterios para determinar las pruebas estáticas que debe agregar al plan de pruebas.
- OBJ7: Conocer la importancia de las pruebas dinámicas, e identificará los tipos de pruebas existentes y contar con los criterios para determinar cuáles pruebas dinámicas incorporar al plan de pruebas.
- OBJ8: Comprender la técnica de pruebas de caja negra y su proceso de aplicación. Contar con los criterios para identificar los tipos de pruebas de caja negra existentes y determinar la fase del STLC en la cual se debe aplicar.
- OBJ9: Comprender la técnica de pruebas de caja blanca y su proceso de aplicación. Contar con los criterios para identificar los tipos de pruebas de caja blanca existentes y determinará la fase del STLC en la cual se debe aplicar.
- OBJ10: Reconocer la importancia del proceso de pruebas de software, identificando las fases del proceso, su utilidad, alcance y organización.
- OBJ11: Comprender los factores que influyen en las pruebas de software relacionados también con la organización.
- OBJ12: Conocer los beneficios de la implementación de PS en las organizaciones, la relación con los requisitos del software y los tipos de riesgos; con los criterios para implementar el proceso de PS en la organización.
- OBJ13: Conocer los principales modelos de calidad en software y su relación con las pruebas de software.
- OBJ22: Conocer la importancia de la documentación en las pruebas de software y plantillas para la documentación de entregables del proceso de planificación de pruebas tales como: política de pruebas de software, estrategia de pruebas, plan de pruebas y los resultados, diferenciando el contenido de cada uno de los elementos en la plantilla y diligenciándolos a partir del enunciado de un proyecto planteado.

tabla 12: Objetivos de Aprendizaje que aportan al logro de las competencias definidas.

OBJ	C1	C2	C3	C4	C5	C6	C7	C10
OBJ1	x							
OBJ2		x						
OBJ3	x							
OBJ4							x	
OBJ5							x	
OBJ6							x	
OBJ7							x	
OBJ8							x	
OBJ9							x	
OBJ10					x	x		
OBJ11					x			x
OBJ12	x		x					
OBJ13					x			
OBJ22				x				

Fuente: Elaboración propia.

### 3. Definición de una estrategia didáctica en las metodologías activas

Estrategia didáctica: Cultura MAKER

Es una metodología activa, que no requiere una instrucción directa, está centrada en el estudiante, y se sustenta en el constructivismo. El estudiante aprende haciendo, siendo el protagonista de su propio aprendizaje. Esta estrategia conjuga la importancia del proceso de construcción paso a paso [29] y cuestionamiento del aprendizaje activo [30].

En la cultura Maker el docente busca que el estudiante aprenda a partir de la exploración. Lo cual potencia las condiciones en un escenario individual de aprendizaje, donde el proceso formativo se convierte en un proceso de apoyo al aprendizaje activo que genera el estudiante.

El aprendizaje derivado de la estrategia MAKER propicia las habilidades y competencias demandas por la industria y la academia en el contexto de la industria 4.0 como: la co-creación, la creatividad, el pensamiento crítico entre otros [31], [32], [33].

Para la implementación de la estrategia se definen artefactos didácticos tales como el plan didáctico de pruebas (PDP), las lecciones de aprendizaje, test de Conocimiento, laboratorio de pruebas, estudio de casos, formatos de documentos de la prueba, como apoyo en los subprocesos del aprendizaje individual anteriormente descritos.

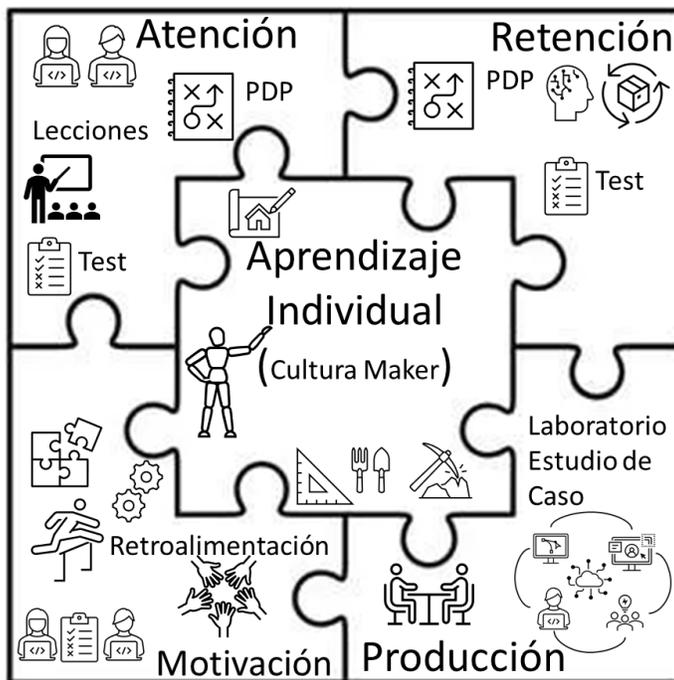


Figura 6: Artefactos didácticos planteados para la implementación de la estrategia.  
Fuente: Elaboración propia.

### 4. Desarrollo de entorno de enseñanza aprendizaje digital

Se construye una aplicación que soporta el proceso de enseñanza aprendizaje de las pruebas de software basado en la norma ISO 29119. Esta aplicación se convierte en un ambiente para el desarrollo de habilidades y competencias requeridas en el proceso formativo propuesto de las pruebas de software en la educación superior y para la implementación de los artefactos didácticos planteados como soporte a la estrategia maker.

El Software asistente en el proceso de pruebas denominado PRUEBASOFTUDES-TDEA contempla varios módulos para el desarrollo de lecciones de aprendizaje y la documentación de pruebas de software, entre estos se encuentra el Módulo gestor de lecciones de aprendizaje, el gestor de competencias, gestor de grupos, gestor de pruebas de conocimiento, laboratorio y módulo de reportes.

El Software permite la configuración de lecciones por parte de un docente, la asignación de lecciones a grupos de estudiantes, el seguimiento al desarrollo de las lecciones y metas del estudiante. Las lecciones se diseñan en función de metas u objetivos de aprendizaje en el marco de una competencia de formación.

La lección maneja una secuencia didáctica determinada por tres momentos: (i) Primer momento corresponde a una conceptualización a la cual se le ha denominado Aprende. El Segundo momento a un proceso de control de aprendizaje, denominado Practica, el cual obedece a un test de conocimiento. El tercer momento corresponde a una práctica del proceso de prueba de Software basado en el uso de estudios de caso, o en el desarrollo de un caso de prueba real, mediante el uso del laboratorio de prueba, al cual se le ha denominado Aplica. Los estudios de casos son estructurados por el docente a través de una plantilla suministrada en el Software. En el laboratorio el estudiante podrá desarrollar la aplicación del plan didáctico de pruebas (PDP) para el software. La Figura 6. Representa la conexión de los módulos diseñados en el software con los diferentes momentos del aprendizaje del estudiante y los artefactos didácticos planteados

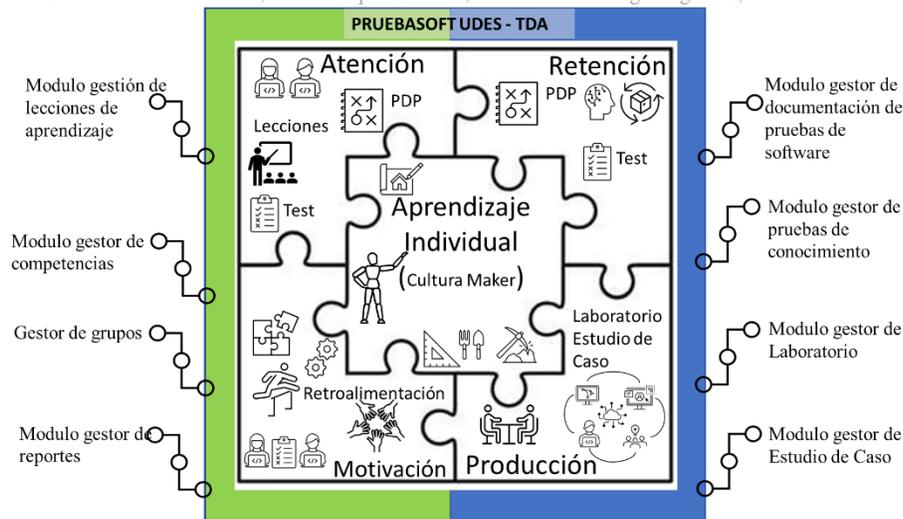


Figura 7: Conexión Módulos momentos del aprendizaje y artefactos didácticos.  
Fuente: Elaboración propia.

## 5. Validación del modelo

La validación se realizó en 4 momentos: M1. Validación de competencias, se inicia esta fase con la validación de las competencias definidas para el modelo, para ello se asignan 4 expertos en el área, los cuales son los encargados de analizar las competencias y lecciones asociadas a cada una y determinar el grado de aceptabilidad de las mismas; M2: Validación del software asistente – PRUEBASOFT UDES-TDEA, en este momento se realizó el registro de los asistentes en el aplicativo para el reconocimiento y validación funcional del mismo, cada usuario fue relacionado a un grupo de estudio y se desarrollaron las lecciones asociadas a las competencias asignadas para cada uno; M3. Pruebas de aceptación, los asistentes diligenciaron los formatos de pruebas de aceptación con el fin de determinar si el software desarrollado es lo que los usuarios esperaban luego de la socialización del modelo; M4. Encuesta de satisfacción, finalmente en el momento tres se compartió un mecanismo tipo encuesta con el propósito de recolectar el nivel de satisfacción y percepción de los estudiantes y docentes con respecto a la implementación de las estrategias formuladas en el modelo, la cual es aplicada en el ámbito académico específicamente en instituciones de educación superior. A continuación, se relacionan los resultados obtenidos de la fase de validación:

### • M1: Validación de competencias.

El objetivo de la validación de las lecciones fue conocer las posibles falencias y atributos positivos respecto al contenido de esta. La lección es una columna importante en el proyecto porque es el contenido que los estudiantes van a aprender y se quiere que con el software su aprendizaje sea alto para poder cumplir con los objetivos del proyecto.

• Validación del software asistente – PRUEBASOFT UDES-TDEA. Se inscribieron al aplicativo un total de 20 personas entre estudiantes y docentes, para la validación de la estrategia se dividieron en 3 grupos, a los cuales se les asignó una competencia diferente de acuerdo con el nivel de estudios y experiencia de cada participante. El ejercicio consistió en verificar las competencias asignadas y realizar las lecciones asociadas de acuerdo con los pasos que sugiere el aplicativo.

### • M2: Pruebas de aceptación.

De acuerdo con el análisis de los formatos de las pruebas de aceptación realizadas con los grupos previamente planeados, se concluyó que el software cumple con sus objetivos agregando unas mejorar en la parte visual y clara para los estudiantes.

### • M3: Encuesta de satisfacción.

Una vez culminado los momentos M2 y M3, los participantes en la sesión de validación contestaron el instrumento formulado para obtener el nivel de percepción del modelo con respecto a su utilización en el entorno educativo.

## V. CONCLUSIONES

La identificación de las competencias relacionadas a profesionales dedicados a las pruebas de software es esencial para el desarrollo estrategias educativas orientadas a la cualificación de estudiantes de las áreas afines a la ingeniería de software, ya que posibilita una adecuada apropiación de los roles y funciones en un entorno empresarial en una organización.

La implementación del proceso de enseñanza de PRUEBAS DE SOFTWARE en un espacio con la infraestructura adecuada permite elevar la calidad de los productos obtenidos de los estudiantes y el desarrollo de habilidades necesarias para el entorno productivo.

Una necesidad identificada en el sector del software supone el mejoramiento de los conocimientos y habilidades de los profesionales en pruebas de software dado que son reducidas actualmente, teniendo en cuenta el déficit de profesionales en la industria TIC.

## VI. REFERENCIAS

- [1] F. A. Vargas-Agudelo, D. E. Soto-Duran, J. C. GiraldoMejía, and C. E. Durango-Vanegas, «Representación en el Núcleo de SEMAT de la Norma ISO/IEC/IEEE 29119-2 para identificar patrones en Pruebas de Software,» in Memoria 4to. Congreso Internacional de Investigación e Innovación en Ingeniería de Software, CONISOFT, pp. 41–46, 2016.
- [2] D. E. Soto-Duran, J. A. Jimenez-Builes y A. X. Reyes-Gamboa, «A knowledge management model for improving the software test process.,» de In Proceedgins of the 18th European Conference on Knowledge Management, Barcelona, 2017.
- [3] S. Tobon, «Formación basada en competencias,» 2006.
- [4] L. Schwartz, «Informe de la tercera reunión de la Comisión de la Unión Europea,» Paris, 1994.
- [5] (25 de julio de 2019). Decreto 1330.
- [6] O. Valverde, «El enfoque de la competencia laboral.,» de Departamento de Publicaciones de Cinterfor/OIT. , Montevideo, 2001.
- [7] G. L. Boterf, «Ingeniería de las competencias,» Barcelona, 2001.
- [8] J. Gozález y R. Wagenaar, «Tuning Educational Structures in Europe,» Universidad de Deusto, Bilbao, 2003.
- [9] M. Escalona, V. Tanja E. y J. Gutierrez, «Pruebas de software en la enseñanza universitaria de la informática: un título propio,» Jornadas de Enseñanza de la Informática, pp. 409-412, 2012.
- [10] X. Luo y M. Störmer, «Chancen und Herausforderungen der Organisations-und Personalentwicklung im Zeitalter der Industrie 4.0- Bestandsaufnahme und Ausblick. In Kommunikation und Technik.,» Springer VS, pp. 191-209, 2018.
- [11] S. Coşkun, Y. Kayıkçı y E. Gençay, «Adapting Engineering Education to Industry 4.0 Vision.,» Technologies, vol. 7, n° 1, 2019.
- [12] R. M. Castro y E. S. Siqueira, «Alternative teaching techniques (active learning) for computer disciplines: A systematic mapping in the Brazil context,» Proc. School Comput. Sci. Workshop, vol. 25, n° 1, 2019.
- [13] I. C. Ribeiro M. y O. M. Passos, «A Study on the active methodologies applied to teaching and learning process in the computing area, » IEEE Access, 2020.
- [14] D. Kolb, *Experiential learning: Experience as the source of learning and development*, Englewood Cliffs: NJ: Prentice-Hall, 1984.
- [15] E. M. Espinar Álava y J. A. Viguera Moreno, «El aprendizaje experiencial y su impacto en la educación actual,» Revista Cubana de Educación Superior, 2020.
- [16] G. Maturro Mazoni, «Modelo para la gestión del conocimiento y la experiencia integrada a las prácticas y procesos de desarrollo software (Doctoral dissertation, Informatica),» 2010.
- [17] D. I. Castañeda y A. M. Pérez-Acosta, «Cómo ocurre el aprendizaje individual en el aprendizaje organizacional? Una revisión del modelo Crossan, Lane y White.,» Revista Interamericana de Psicología Ocupacional, vol. 24, pp. 1-15, 2005.
- [18] A. Bandura, «Teoría del Aprendizaje Social,» Espasa-Calpe, 1982.
- [19] A. Bandura, *Social foundations of thought and action: A social cognitive theory.*, New Jersey: Prentice Hall, 1986.
- [20] D. E. Soto-Duran, «Tesis Doctoral Modelo de gestión de conocimiento aplicado a las pruebas de software.,» 2017.
- [21] J. Aquilino A., J. d. Andrés, C. Nieto, M. Suárez, J. R. Pérez, A. Cernuda, C. Luendo, A. B. Martínez, M. Riesco, D. F. Lanvín, J. E. Labra, M. D. Fondón y J. M. Redondo, «Definición de competencias específicas y genéricas del ingeniero en informática,» Docencia Universitaria Proyectos de Innovación Docente., 2006.
- [22] IEEE Computer Society, «Software Engineering Competency Model Version 1.0 (SWECOM),» 2014.
- [23] P. Bourque y R. Fairley, «Guide to the Software Engineering Body of Knowledge, Versión 3.0, IEEE Computer Society, .,» 2014.
- [24] ACM/IEEE Computer Society, «Computer Engineering Curricula 2016,» 2016.
- [25] A. Flebes, T. Capote, Y. León, A. Velásquez, R. Delgado y R. Calzadilla, «Una experiencia novedosa para el testing desarrollada por un departamento de pruebas de software.,» Revista cubana de ciencias informáticas, pp. 1-15, 2011.
- [26] W. Sheikh, «Teaching C++ programming using automated unit testing and test-driven development—Design and efficacy study,» Computer Applications in Engineering Education, vol. 30, n° 3, pp. 821-851, 2022.
- [27] A. Cernuda, «Una experiencia de generación personalizada de ejercicios de pruebas de software.,» Jornadas de enseñanza universitaria de la informática (JENUI), pp. 190-196, 2010.
- [28] D. Jústiz, D. Gómez y M. Delgado, «Proceso de pruebas para productos de software en un laboratorio de calidad.,» Ingeniería industrial, pp. 131-145, 2014.
- [29] D. Dougherty, «The Maker Movement,» Innovations: Technology, Governance, Globalization, vol. 7, n° 3, pp. 11-14, 2012.
- [30] J. Dewey, *Experience and Education*, New York: The Macmillan Company, 1956.
- [31] L. Martín, «The Promise of the Maker Movement for Education,» Journal of Pre-College Engineering Education Research (J-PEER), vol. 5, n° 1, pp. 30-39, 2015.
- [32] B. Taulor, «Evaluación del beneficio del movimiento maker en la educación K-12 STEM,» Electronic International Journal of Education, 2016.
- [33] M. Montanero Fernandez, «Métodos pedagógicos emergentes para un nuevo siglo ¿qué hay realmente de innovación?,» Revista Interuniversitaria, pp. 5-34, 2019.