

# Proceso de pruebas de *software* para un modelo de calidad en Cuba

## Process of *software* testing for a model of quality in Cuba

Lisandra Díaz Figueredo <sup>1\*</sup>, Yoandy Lazo Alvarado <sup>2</sup>, Leanet Tamayo Oro <sup>2</sup>

<sup>1</sup> Empresa de Tecnologías de la Información (ETI), Biocubafarma, Cuba

<sup>2</sup> Dirección de Consultoría y Evaluación a Procesos, Centro Nacional de Calidad de Software (CALISOFT), Cuba

\*Autor de correspondencia: [lisandra@eti.biocubafarma.cu](mailto:lisandra@eti.biocubafarma.cu)

**RESUMEN**— La industria de *software* se desarrolla a un ritmo acelerado y la competitividad en el mercado exige a las empresas alta calidad en los procesos y productos. La calidad se obtiene al realizar una correcta gestión, aplicando actividades de planificación, aseguramiento y control de la calidad y mejoras. Los modelos de calidad proponen un conjunto de buenas prácticas enfocadas a los procesos de gestión. Cuba cuenta con un modelo aplicable al país, basado en las buenas prácticas de los modelos y normas de referencia, nombrado Modelo de Calidad para el Desarrollo de Aplicaciones Informáticas (MCDAI). En el caso de las actividades de control de la calidad sobresalen los procesos de Verificación y Validación que aseguran el cumplimiento de los requisitos de las partes interesadas; las pruebas de *software* es la principal técnica dinámica de estos procesos. Esta investigación propone un conjunto de requisitos específicos que sintetizan las buenas prácticas de pruebas de *software* en un proceso base. Fue diseñado para formar parte de los procesos base del MCDAI. Incluye una descripción gráfica y textual del proceso, que facilita el cumplimiento de los requisitos, y los roles involucrados. Este proceso base fue sometido a grupos focales y a encuesta de satisfacción de expertos para comprobar la utilidad de la propuesta y a proyectos piloto para verificar que con su aplicación aumenta la efectividad de las pruebas en etapas tempranas durante el desarrollo de *software*.

**Palabras clave**—Prueba de *software*, modelo de la calidad, verificación, validación.

**ABSTRACT**— The software industry is developing at an accelerated pace, and the competitiveness in the market requires company's high-quality processes and products. Quality is obtained by performing proper management, applying planning activities, quality assurance and control and improvements. The quality models propose a set of good practices focused on management processes. Cuba has a model applicable to the country, based on the best practices of the reference models and standards, named Quality Model for the Development of Computer Applications (MCDAI). In the case of quality control activities, the Verification and Validation processes ensure compliance with stakeholder requirements; software testing is the main dynamic technique of these processes. This research proposes a set of specific requirements that synthesize good software testing practices in a base process. It was designed to be part of the base processes of the MCDAI. It includes a graphic and textual description of the process, which facilitates compliance with the requirements and the roles involved. An indicator system is defined to provide the project and the management of an organization with an objective view of the processes and products of associated works. This process was submitted to focus groups, and an expert satisfaction survey to verify the usefulness of the proposal and to pilot projects to verify that with its application the effectiveness of the tests increases during software development.

**Keywords**— *Testing of software, quality model, verification, validation.*

## 1. Introducción

La industria de *software* se desarrolla a un ritmo acelerado y la competitividad en el mercado exige a las empresas alta calidad en los procesos y productos que desarrolla. Según Pressman, la calidad del *software* se obtiene cuando se usan prácticas y procesos probados de la Ingeniería del *Software*, cuando se planifican y

gestionan correctamente los proyectos, se realiza el control exhaustivo y se tiene una infraestructura de aseguramiento de la calidad [1], [2], [3]. De este planteamiento se concluye que cuando se gestiona la calidad, se logra la calidad del software.

Para asegurar los requisitos del usuario y la calidad esperada de un producto de software, son fundamentales

los procesos de Verificación y Validación (V&V) [4], [5]; el primero establece la correspondencia entre un producto y su especificación, y el segundo, define si un producto de software es adecuado para su uso en el entorno operativo previsto [6], [7], [8]. Las pruebas de software es la principal técnica dinámica de estos procesos, implican ejecutar una implementación del *software* con datos de prueba, se examinan las salidas y su entorno operacional para comprobar que funciona tal y como se requiere [9].

Al analizar los conceptos definidos por [10], [11], [12], [13], [14], [15] se concluye que las pruebas de software constituyen el conjunto de actividades realizadas para facilitar el descubrimiento y/o la evaluación de propiedades de uno o más elementos de prueba. Se considera un proceso crítico que puede demostrar la presencia de defectos, pero nunca demostrar la ausencia de ellos. Puede ser usado para prevenir los defectos, determinar la información necesaria y el nivel de calidad para la toma de decisiones.

Cuba desarrolla acciones para fortalecer la Industria Cubana de Programas y Aplicaciones Informáticas, entre las que se encuentra el Modelo de la Calidad para el Desarrollo de Aplicaciones Informáticas (MCDAI) [16], y las Reglas básicas para la producción de *software* [17] El MCDAI tiene un enfoque a procesos y es la propuesta aceptada para la Industria Cubana de Programas y Aplicaciones Informáticas (ICPAI) [18].

Un diagnóstico realizado en el año 2014 por el Centro Nacional de Calidad de Software (CALISOFT) a una muestra del 43.75% de las organizaciones desarrolladoras de *software* en Cuba, permitió identificar que: el 57% son pequeñas y medianas empresas (PYME) [19] [18], el 55.74% aplican metodologías ágiles y 85.7% utilizan un enfoque a proceso [20].

Los resultados del diagnóstico también permitieron identificar que el 28.57% de las organizaciones no implementa un proceso de pruebas de *software*, el 78.2% no realiza pruebas basadas en los requisitos funcionales y no funcionales, el 65% no diseña casos de prueba, el 55.1% ejecuta las pruebas en el ambiente de producción, obviando el entorno controlado que se prepara para ejecutar las mismas, y el 16.42% no cumple la totalidad de los requisitos pactados con el cliente [20] [21].

En el 2017 dicha entidad, aplicó otro diagnóstico a una muestra del 28.13% de ICPAI. Los resultados permitieron corroborar la situación en el desarrollo de *software* del país respecto a las pruebas de *software*, sobresaliendo que el 55% de las organizaciones no define un plan de prueba; el 65% no realiza pruebas unitarias y de aceptación; el 45 % no realiza la gestión de los defectos; el 50.3% no prepara los entornos de prueba; y el 89.3% no realiza acciones para prevenir no conformidades [22].

En un análisis realizado por el Departamento de Evaluación de Productos, perteneciente a CALISOFT en el periodo 2014 al 2017, demostró que el 70% de los proyectos que solicitaron las evaluaciones a productos no entregaron evidencia del diseño de los casos de prueba, ni el de la gestión de defectos. En las pruebas que realiza dicha entidad, se obtuvo que el 70% de las no conformidades identificadas en el entorno operacional pudieron ser detectadas y corregidas en etapas tempranas. Este resultado demuestra que las pruebas de *software* se ejecutan incorrectamente o no se hacen durante el desarrollo.

A partir de la situación problemática antes expuesta, se plantea como objetivo principal de esta investigación: elaborar un proceso base de Pruebas de *software* para el MCDAI que permita aumentar la efectividad de las pruebas durante el proceso de desarrollo de *software*.

Según la Real Academia de la Lengua Española, la efectividad es la capacidad de lograr el efecto que se desea o se espera [23]. A los efectos de esta investigación la efectividad es la capacidad de lograr que aumenten la detección de los defectos en etapas tempranas durante el desarrollo de *software* y disminuyan las No conformidades (NC) en el entorno operacional.

## 2. Materiales y métodos

Para cumplir con el objetivo planteado, se realizó una revisión bibliográfica sobre los procesos de V&V y pruebas de *software* en modelos de la calidad y normas internacionales, como son: ISO/IEC/IEEE 12207:2017 [8], CMMI-DEV [24], COMPETISOFT [25], MoProSoft [26], ISO/IEC/IEEE 29119-1:2013 [27], TMMI [28], ISO/IEC/IEEE 29119-2:2013 [10] y *Software Qualification Testing Process* [15]. Además se consultó el diseño y los resultados de investigaciones científicas

realizadas con anterioridad [2], [29], [30], [31] y la literatura [13], [12], [1], [11], [9]. La Tabla 1, muestra un resumen de este estudio, la cual constituye la base de esta propuesta.

El análisis evidenció que las buenas prácticas de pruebas de *software*, se tratan de manera general y están incluidas como parte aislada de otros procesos. En todos se analizan y diseñan casos de prueba, preparan entorno y ejecutan las pruebas; sin embargo, solo ISTQB e ISO/IEC/IEEE 29119-2 poseen un proceso específico de pruebas. CMMI, MPS.Br e ISO/IEC/IEEE 12207 incluyen los procesos de Validación y Verificación. La estrategia de pruebas en la mayoría de los casos se encuentra dentro de la planificación del proyecto y se obvia información importante para la toma de decisiones. En el caso de determinar cobertura y evaluar las características de la calidad, solo CMMI las incluye dentro del conjunto común de medidas de proceso y de producto para los procesos estándar de la organización, y ISTQB e ISO/IEC/IEEE 29119-2 en la fase de diseño. En el caso de analizar y diseñar las pruebas para la reutilización, se emplea el término: reutilización de los casos de prueba. Dentro del análisis de los resultados en CMMI se incluye identificar las causas de los defectos, mientras que en el modelo de pruebas TMMI se encuentran en el nivel optimizado, correspondiente al área de proceso: Prevención de defectos. El seguimiento y control para el proyecto de pruebas se tiene en cuenta en ISTQB como control, y ISO/IEC/IEEE 29119-2 y TMMI como el Proceso de control y monitoreo de pruebas, porque son específicos de pruebas, pero en el caso de los demás, ocupa lugar en otras áreas o procesos.

Teniendo en cuenta lo antes descrito, donde las actividades de prueba se realizan de forma generalizada, y la mayoría forman parte de los procesos de V&V lo cual limita su especialización y los recursos (tiempo y personal); se decidió realizar una propuesta propia de requisitos que indiquen “*qué*” hacer para ejecutar las pruebas al producto de *software*, incorporando paulatinamente buenas prácticas para su gestión y ejemplificando “*cómo*” ponerlos en práctica mediante un proceso que contenga actividades, roles asociados, técnicas y herramientas.

## 2.1 Modelo de Calidad para el Desarrollo de Aplicaciones Informáticas

El MCDAI tiene como objetivo proporcionar a las entidades desarrolladoras de *software* un modelo sustentado en las mejores prácticas teniendo en cuenta las características nacionales y basándose en los siguientes principios (fácil de entender, ligero y que sirva de base para alcanzar evaluaciones en otros modelos o estándares). El modelo está dirigido a entidades dedicadas al desarrollo y/o mantenimiento de *software*. Está enfocado a las Pymes por ser mayoría en la Industria Cubana de Programas y Aplicaciones Informáticas, aunque también puede ser usado por grandes empresas. Cualquier organización que no cuente con procesos establecidos puede usar el modelo ajustándolo de acuerdo con sus necesidades, mientras que las que ya tienen procesos establecidos pueden usarlo como punto de referencia para identificar los elementos que pueden mejorar [18]. El MCDAI se compone de: 1) Guía general, 2) Guía de implementación y 3) Guía de evaluación. La Guía implementación agrupa los procesos en las categorías: I) Gestión organizacional, II) Gestión de proyectos, III) Ingeniería y IV) Soporte.

**Tabla 1.** Buenas prácticas seleccionadas de modelos y estándares de calidad.  
Fuente: Elaboración propia.

Buenas prácticas	CMMI-DEV	TMMI	MPS.Br	MoProSoft	COMPETISOFT	ISTQB	ISO/IEC /IEEE	
							12207	29119
Elaborar estrategia de pruebas.	x	x	x			x	x	x
Determinar la cobertura de las pruebas.	x	x				x		x
Seleccionar las medidas de la calidad.	x	x				x	x	x
Automatizar la ejecución de las pruebas.	x	x				x		x
Analizar y diseñar las pruebas.	x	x	x	x	x	x	x	x
Determinar los elementos de cobertura.	x	x				x	x	x
Analizar y diseñar las pruebas para la reutilización.	x	x				x	x	x
Configurar entorno de pruebas.	x	x	x	x	x	x	x	x
Ejecutar pruebas.	x	x	x	x	x	x	x	x

Analizar los resultados de las pruebas.	x	x	x	x	x	x	x	x
Identificar las causas de los defectos.	x	x						
Evaluar las características de la calidad.	x	x				x	x	x
Realizar seguimiento y control del proyecto de prueba.		x	x	x	x	x	x	
Finalizar el proyecto de prueba.	x	x	x	x	x	x	x	x

La categoría de Ingeniería reúne los procesos base técnicos necesarios para el desarrollo del *software*. Cada proceso base contiene un propósito, requisitos específicos y la modelación del proceso. En el caso de los requisitos específicos, se definen por niveles Básico, Intermedio y Avanzado; y se dividen en tres partes: título, descripción y evidencia recomendada [18].

### 3. Proceso base de Pruebas de software

#### 3.1 Propósito y requisitos específicos

Esta investigación propone el proceso base de Pruebas de *software* y como es parte del MCDAI está alineado con su estructura. Tiene el **propósito** de comprobar que se haya elaborado el producto o componentes de productos de acuerdo con lo especificado y que se hayan cumplido los requisitos: para una utilización o aplicación específica, prevista por las partes interesadas pertinentes.

Para cumplir con este propósito y en base a las buenas prácticas de pruebas identificadas como parte de la construcción del marco teórico, se propusieron requisitos específicos divididos en tres niveles de madurez. La división de requisitos en los niveles de madurez permite que la propuesta pueda ser adoptada de forma escalonada, incorporando pequeñas mejoras al proyecto.

Tabla 2. Requisitos específicos del proceso base de Pruebas de software

<b>Nivel Básico</b>	PS 3 Analizar y diseñar las pruebas. PS 4 Configurar el entorno de pruebas. PS 5 Ejecutar prueba. PS 5.3 Ejecutar prueba del sistema. PS 5.4 Ejecutar prueba de aceptación. PS 5.5 Ejecutar pruebas de repetición. PS 6 Analizar los resultados de las pruebas. PS 9 Finalizar el proyecto de prueba.
---------------------	--

<b>Nivel Intermedio</b>	PS 1 Elaborar estrategia de pruebas. PS 5.1 Ejecutar pruebas de unidad/componentes. PS 5.2 Ejecutar prueba de integración. PS 5.6 Ejecutar pruebas de regresión. PS 8 Realizar seguimiento y control del proyecto de prueba.
<b>Nivel Avanzado</b>	PS 2 Automatizar la ejecución de las pruebas. PS 3.1 Determinar los elementos de cobertura de prueba. PS 3.2 Analizar y diseñar pruebas para la reutilización. PS 6.1 Identificar las causas de los defectos. PS 7 Evaluar las características de la calidad.

#### 3.2 Proceso y actividades

El proceso base está formado por siete subprocesos, como se muestra en la Figura 1, uno relacionado con la planificación, cuatro con la ejecución de los niveles de prueba, uno con finalizar el proyecto de prueba y el último dedicado al seguimiento y control.

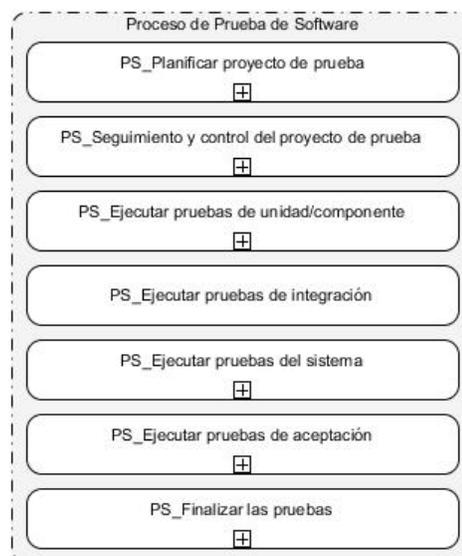


Figura 1. Relación de subprocesos que componen el proceso base Pruebas de software.

#### 3.2.1 Relación del proceso base Pruebas de software con Desarrollo de la solución

El proceso base de Pruebas de *Software* posee estrecha relación con el proceso base de Desarrollo de la Solución del modelo, específicamente con el proceso de DS\_Construir y entregar el producto, el cual se encarga de invocar mediante eventos a los subprocesos de ejecución de las pruebas (unidad/componente, de integración, de sistema y de aceptación) así como

resolver los defectos, no conformidades y peticiones de cambios aceptadas como se muestra en la figura 2.

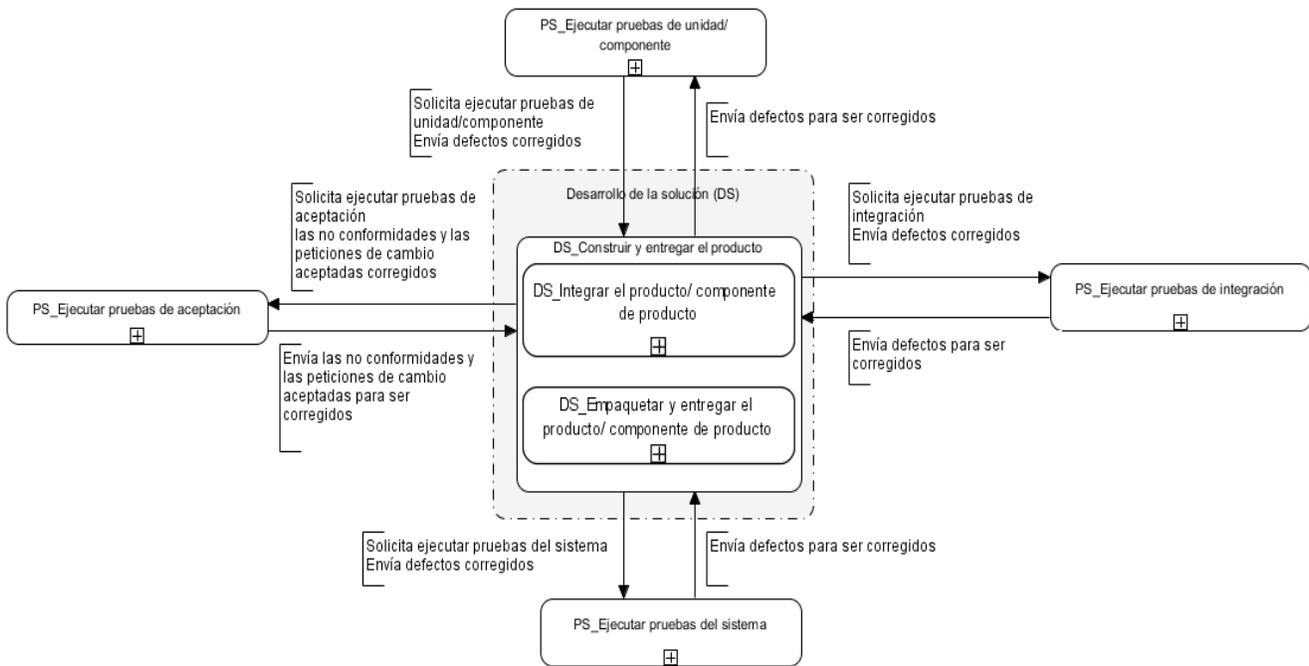


Figura 2. Relación de los procesos base Pruebas de software y Desarrollo de la Solución.

### 3.2.2 Subproceso Planificar proyecto de prueba

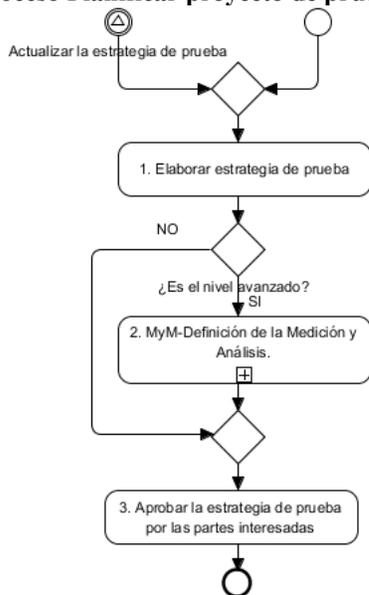


Figura 3. Descripción gráfica del subproceso PS\_Planificar proyecto de prueba.

El subproceso Planificar proyecto de pruebas, como se aprecia en la figura 3, consiste en elaborar una estrategia de prueba siguiendo el enfoque de prueba basado en riesgos. En dependencia de las características de la calidad, se identifican los indicadores para cubrir los requisitos de la calidad del producto y/o componente de producto, y los criterios de decisión para la medida de la calidad y de la evaluación, guiado por el subproceso MyM-Definición de la Medición y Análisis. También se aprueba la estrategia por las partes interesadas y realizan los cambios pertinentes hasta llegar a la aprobación final.

### 3.2.3 Subproceso Ejecutar pruebas de Unidad o componente

El sub proceso PS\_Ejecutar Pruebas de unidad/componente, como se aprecia en la fFigura 5, consiste en instalar y/o configurar en el entorno de desarrollo las herramientas necesarias para diseñar y ejecutar las pruebas unitarias; analizar la base de pruebas para entender las unidades funcionales que serán desarrolladas; codificar los casos de prueba y ejecutar el procedimiento de pruebas unitarias; si se identificaron los defectos se ejecutan las pruebas de repetición y regresión para comprobar si los mismos fueron corregidos.

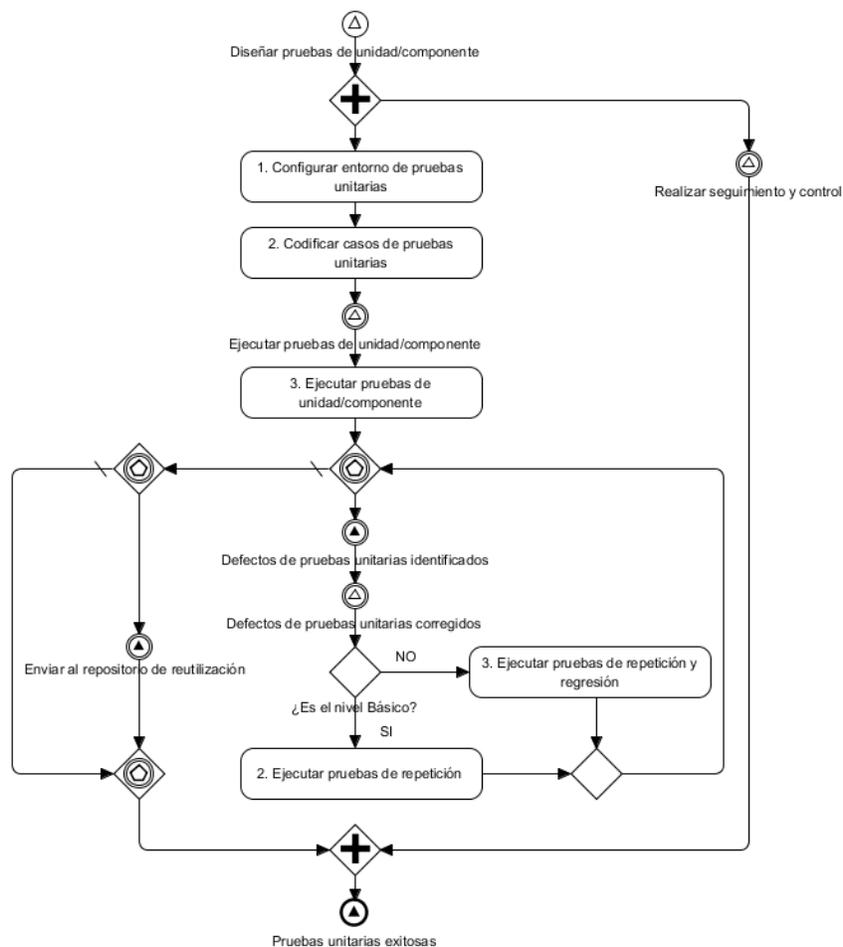


Figura 4. Descripción gráfica del subproceso DPS\_Ejecutar Pruebas de unidad/componente.

### 3.2.4 Subproceso – Ejecutar pruebas de integración

El subproceso PS\_ Ejecutar pruebas de integración como se muestra en la figura 5, consiste en instalar y/o configurar en el entorno de desarrollo las herramientas necesarias para diseñar y ejecutar las pruebas de integración; analizar la Base de prueba para entender las relaciones entre componentes e interfaces, así como los mecanismos de integración y ejecutar el “Procedimiento de pruebas de integración”. Además, teniendo en cuenta los resultados reales de las pruebas compara los resultados esperados con los reales de los casos de prueba, para determinar si pasa o falla la prueba; y si se identificaron los defectos se ejecutan las pruebas de repetición y regresión para comprobar si los mismos fueron corregidos.

### 3.2.5 Subproceso – Ejecutar pruebas del Sistema

El sub proceso PS\_ Ejecutar pruebas del sistema como se muestra en la figura 6 , consiste en analizar la base de pruebas teniendo en cuenta el nivel de prueba, tipo de prueba y características de la calidad a probar definidas; diseñar los casos de prueba teniendo en cuenta los elementos de la base de prueba; crear los servidores virtuales, si son necesarios para instalar el producto y configurar los mismos con las características necesarias para crear el entorno controlado deseado; se ejecutan los Procedimientos de pruebas de sistemas para cada tipo de prueba, según la planificación; se registran los resultados reales obtenidos de cada caso de prueba que compone el procedimiento; compara los resultados esperados con los reales de los casos de prueba, para determinar si pasa o falla la prueba; y en caso de fallo registra y clasifica los defectos.

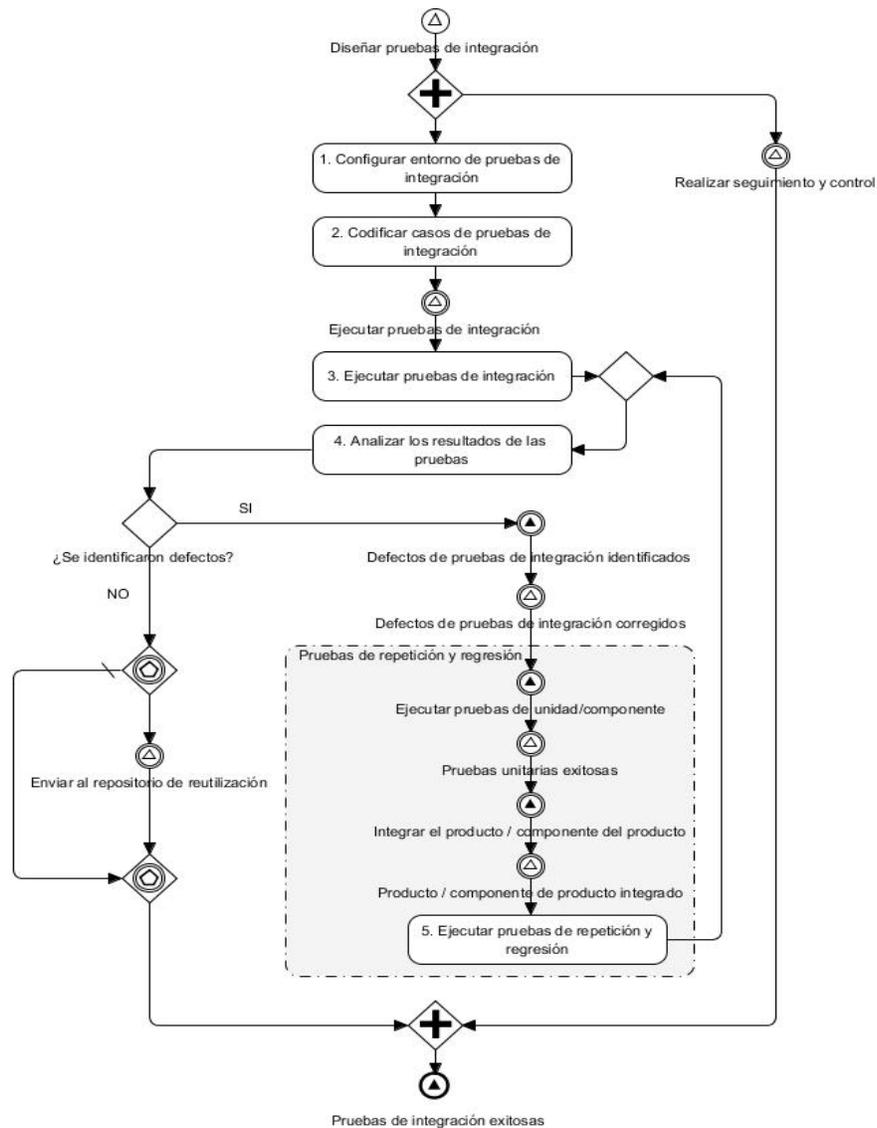


Figura 5. Descripción gráfica del subproceso PS\_Ejecutar Pruebas de integración.

### 3.2.6 Subproceso – Ejecutar pruebas de aceptación

El sub proceso PS\_ Ejecutar pruebas de aceptación como se muestra en la figura 7, consiste en analizar la base de pruebas; identificar los casos de prueba que se requieren diseñar para ejecutar las pruebas; diseñar los casos de prueba teniendo en cuenta los elementos de la base de prueba y aplica la técnica de diseño especificada; crear los servidores virtuales, si son necesarios para instalar el producto en el entorno operacional controlado y configurar los servidores con las características necesarias para crear el entorno operacional controlado deseado. Configurar el entorno en las PC cliente y/o

dispositivos portátiles donde se ejecutarán las pruebas; las partes interesadas pertinentes aceptan los procedimientos de prueba de Aceptación elaborados y ejecutan el mismo, se observan y registran los resultados reales obtenidos de cada caso de prueba. Se identifican si existe conformidad con el producto, pedidos de cambios a los requisitos pactados y/o no conformidades por incumplimiento de estos y si se identificaron no conformidades se ejecutan las pruebas de repetición y regresión para comprobar si las mismas fueron corregidas.

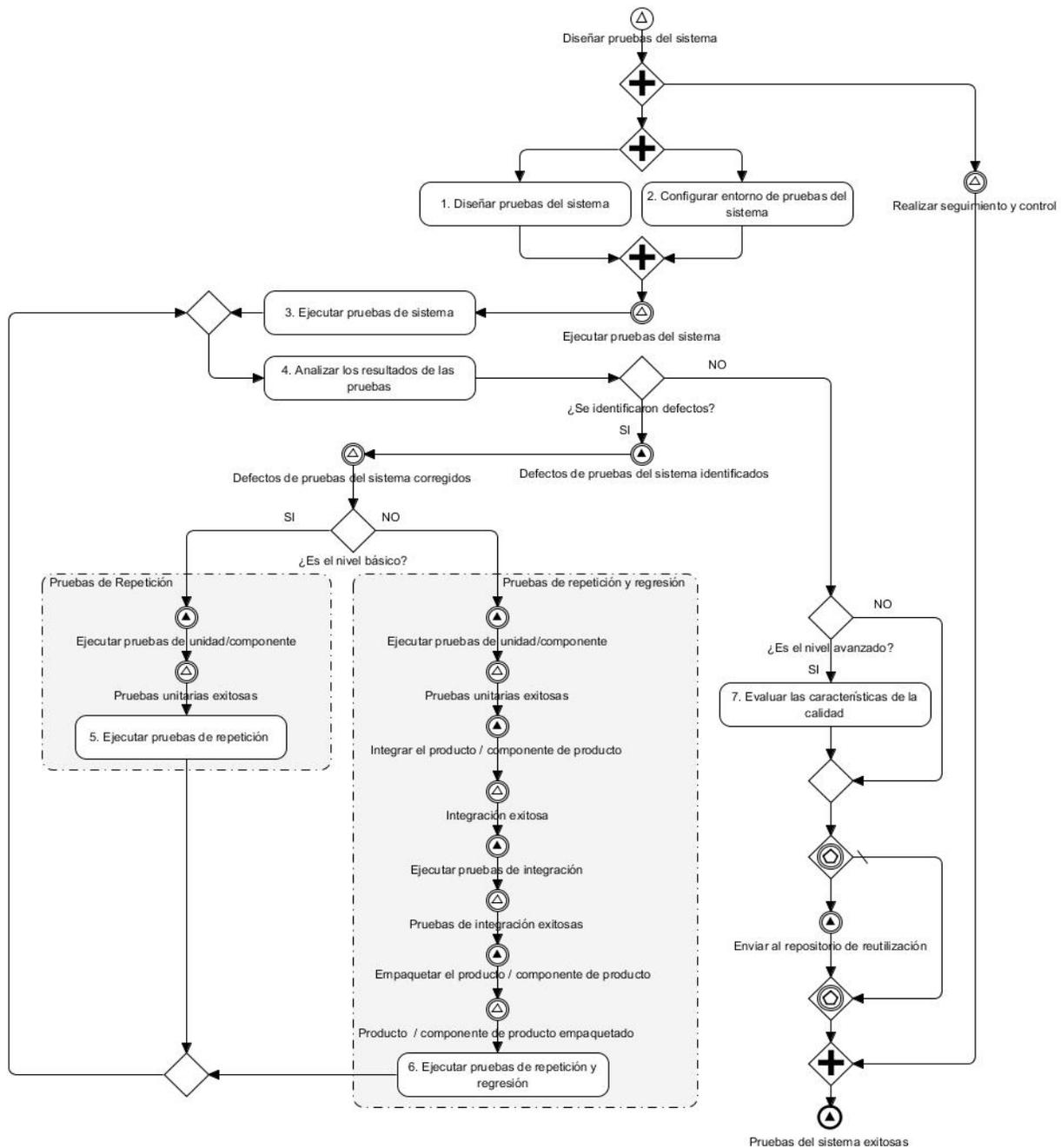


Figura 6. Descripción gráfica del subproceso PS\_Ejecutar Pruebas de sistema.

### 3.2.7 Subproceso PS - Realizar seguimiento y control

El subproceso PS\_ Realizar seguimiento y control como se muestra en la Figura 8, consiste en realizar el seguimiento de la estrategia de prueba relacionada para determinar su cumplimiento y/o desviaciones y analizar

los riesgos existentes del proyecto y producto para identificar cambios en el coeficiente de exposición de los riesgos. Analiza el impacto de las desviaciones en el cumplimiento de la “Estrategia de pruebas” e identifica los problemas asociados.

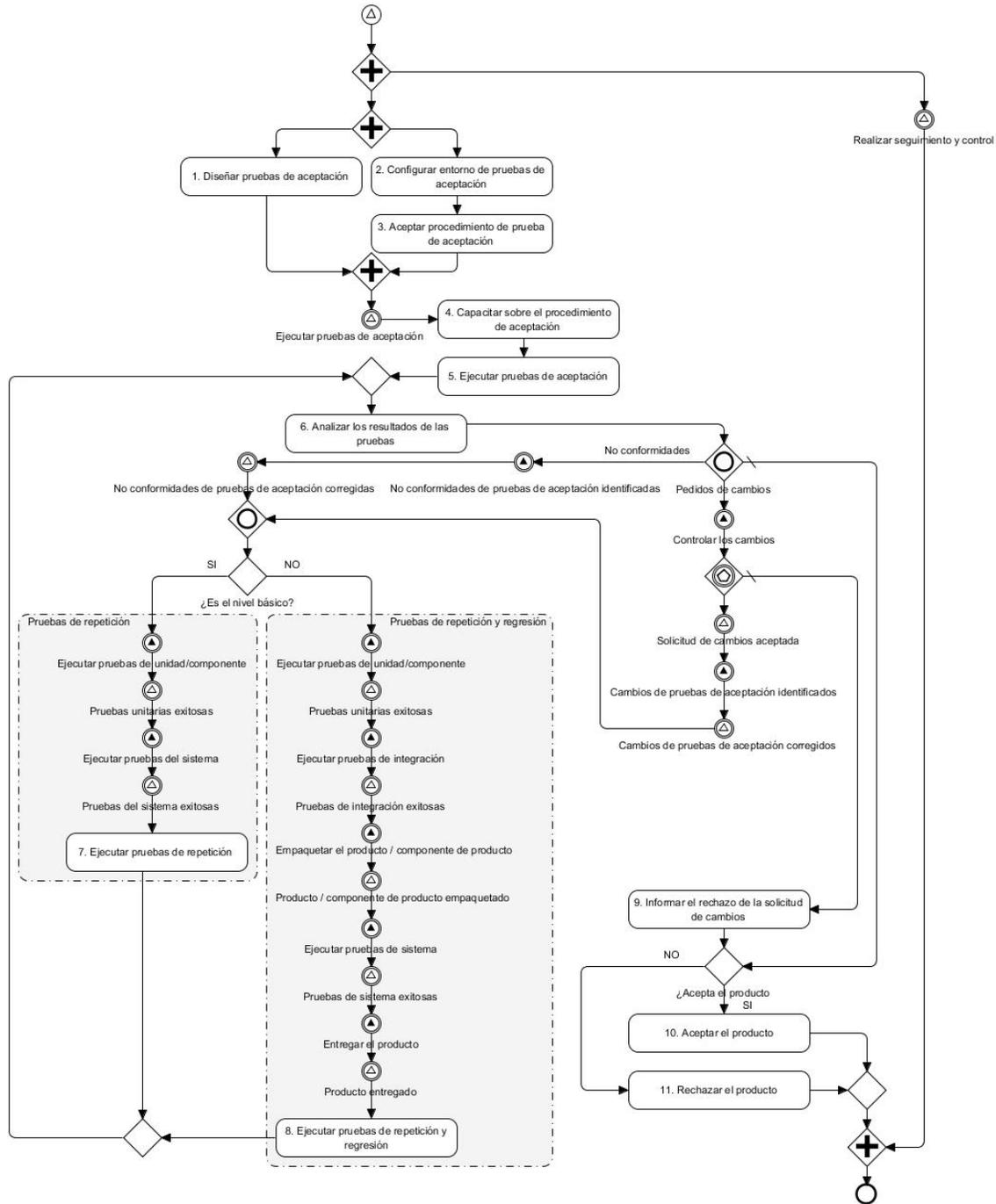


Figura 7. Descripción gráfica del proceso PS – Ejecutar Pruebas de aceptación.

### 3.2.8 Subproceso PS –Finalizar proyecto de prueba

El subproceso PS\_Finalizar proyecto de prueba como se aprecia en la Figura 9, consiste crear un informe que analice el cumplimiento de la planificación del proyecto de prueba, resuma los resultados finales de la

ejecución de las pruebas y que describa los defectos/no conformidades encontradas; por otra parte, se analiza y concilia dicho informe y archivan todos los artefactos generados.

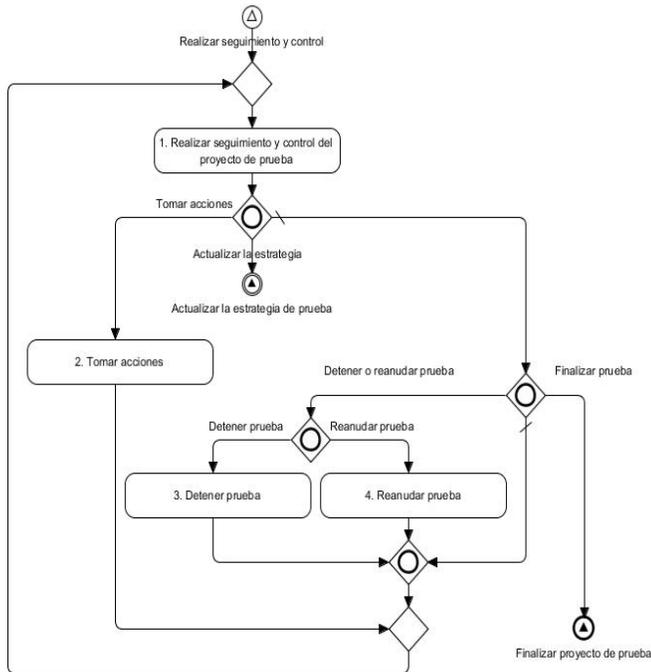


Figura 8. Descripción gráfica del proceso PS - Realizar seguimiento y control.

### 3. Validación

Para validar los requisitos y el proceso base de Pruebas de software, se empleó la técnica de grupos focales; que consiste en la discusión libre y espontánea de determinado tema por grupos pequeños de personas. La discusión es guiada por un moderador y se registran todos los criterios que se emiten [32]. Para su conformación se tuvo en cuenta los criterios emitidos por Aigner [33], al decir que el tamaño del grupo debe oscilar entre cuatro (4) y doce (12) participantes; asegurando que todos puedan emitir sus valoraciones y que haya riqueza de ideas. Además, debe ser homogéneo, determinado por el propósito del estudio [21].

Para cumplir con lo antes expuesto se seleccionaron 10 especialistas en calidad y pruebas de *software* con un promedio de 8 años de experiencia de las entidades: CALISOFT (1), Banco Nacional de Cuba (2), CUJAE (3), GET (2), Xetid (1) y ETECSA (1), los cuales fueron convocados a participar en 7 talleres para debatir la propuesta.

Se tomaron alrededor de 60 acuerdos que se cumplieron en el tiempo planificado y permitieron mejorar el proceso propuesto, logrando así procesos corregidos y refinados

en correspondencia con las particularidades de la industria cubana. En el último encuentro, los profesionales plantearon que el proceso base de pruebas de *software*, cumple con las necesidades que tienen las organizaciones de la industria y su utilidad para adaptarse fácilmente a los diversos entornos de desarrollo de *software*.

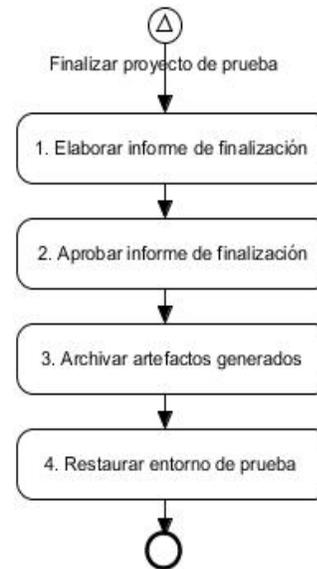


Figura 9. Descripción gráfica del proceso PS - Finalizar proyecto de prueba.

Concluida la propuesta teórica del proceso se utilizó el criterio de expertos, específicamente la escala psicométrica creada por Rensis Likert [34], como método para procesamiento estadístico de criterios. Para ello, fueron seleccionados posibles expertos a los cuales se les aplicó una encuesta para determinar el coeficiente de competencia de estos y se seleccionaron los adecuados, se les presentó una encuesta con la propuesta de los requisitos del proceso base de pruebas de *software*, se procesó y valoró la información obtenida. Los 25 expertos seleccionados son graduados universitarios, con un promedio de 12 años de experiencia en roles de administrador de la calidad, líder de pruebas y probador; 14 máster y 3 doctores; 14 poseen la certificación Tester Foundation Level y 5 Tester Advanced Level, Test Manager. Dichos expertos emitieron sus criterios y evaluaron del 1 al 5 la calidad de los requisitos planteados. Al analizar los resultados se puede concluir que los requisitos PS2, PS4.6 y PS5 obtuvieron la

calificación más alta y que el PS2.2 la más baja, puesto que el 4 % le asignó 3 puntos. Este resultado permitió afirmar que PS2, PS4.6 y PS5 se encontraban listos para utilizar en el proceso y publicar, mientras que el resto con puntuaciones similares fueron modificados. En el caso de PS2.2 “Determinar elementos de cobertura” se le modificó la redacción y se identificó que debe ir antes de PS 2.1 “Seleccionar las medidas de la calidad”; se creó un nuevo requisito y se puso como sub-requisito de PS 1, enfocado a la planificación de la cobertura de prueba y

los criterios de finalización. El índice porcentual relacionado con la valoración de los expertos de cada requisito se evidencia en la figura 10, el cual alcanza o supera el 85.5% en todos los casos lo que demuestra una alta valoración de los expertos respecto a la propuesta teórica del proceso base.

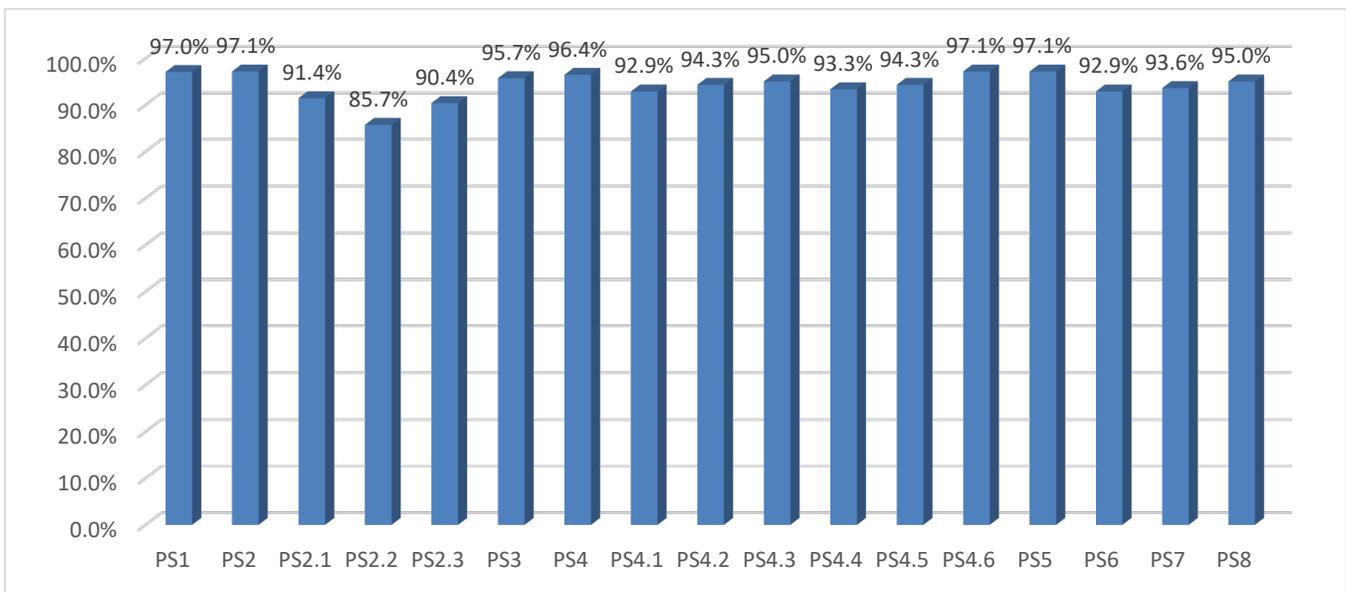


Figura 10. Valoración de los expertos con respecto a la calidad de los requisitos.

Para valorar el efecto de la implementación del proceso, se utilizó el método de preexperimento [35], con el objetivo de demostrar que con el proceso propuesto aumentó la detección de los defectos en etapas tempranas durante el desarrollo de *software* y disminuyeron las NC en el entorno operacional.

El preexperimento se realizó a través del diseño de preprueba - postprueba con un solo grupo. El mismo permite determinar un punto de referencia inicial para ver qué nivel tenía el grupo en las variables dependientes antes del estímulo, permitiendo su seguimiento [34]. Se realizaron en cuatro proyectos piloto correspondientes a al grupo empresarial Biocubafarma, con características similares en estructura y funciones. Los proyectos están formados por siete especialistas: un jefe de proyecto, dos analistas, un probador, dos programadores y un gestor de

la configuración) que desarrollan aplicaciones web utilizando el mismo lenguaje de programación y tecnologías para el desarrollo.

El pre-experimento se realizó en los meses de junio a septiembre del presente año y se obtuvo el resultado que se muestra en la tabla 3.

Tabla 3. Análisis preprueba - postprueba. Efectividad del proceso de pruebas.

Antes (Sin aplicar la propuesta)	Indicadores		Después (Aplicada la propuesta)	Indicadores	
	Cant. Defectos	Cant. NC		Cant. Defectos	Cant. NC
P1	100	20	P1	150	5
P2	125	12	P2	200	10
P3	220	4	P3	80	3
P4	115	75	P4	230	10

A continuación, se especifican los indicadores que fueron evaluados en el preexperimento:

Cant.Defectos = Cantidad de defectos identificados por el equipo de proyecto en etapas tempranas durante el proceso de desarrollo de *software*.

Cant.NC = Cantidad de NC detectadas por el cliente en el entorno operacional del *software*.

Pi = Proyectos del 1 al 4 que aplicaron la propuesta.

Como resultado de la investigación, se pudo apreciar una mejoría en los indicadores de efectividad del proceso luego de aplicada la propuesta (con estímulo), se redujeron las NC en el entorno operacional del *software* y aumentó la detención temprana de defectos por parte del equipo de proyecto; por tanto, se puede concluir que con la aplicación del proceso propuesto se aumenta la efectividad de las pruebas durante el proceso de desarrollo de *software*.

Para conocer la satisfacción de usuarios con la propuesta se empleó la técnica de IADOV [36], [18], [21] y [34] a los especialistas que aplicaron el proceso base en sus proyectos.

En los resultados que se muestran en la figura 11 se puede apreciar que el 80% de los usuarios muestran una clara satisfacción con la propuesta presentada. Además, no se contó con usuarios insatisfechos, por lo que se obtuvo un Índice de Satisfacción Grupal de 0.9, en base a 1, lo que indica la satisfacción de los usuarios con el proceso propuesto.

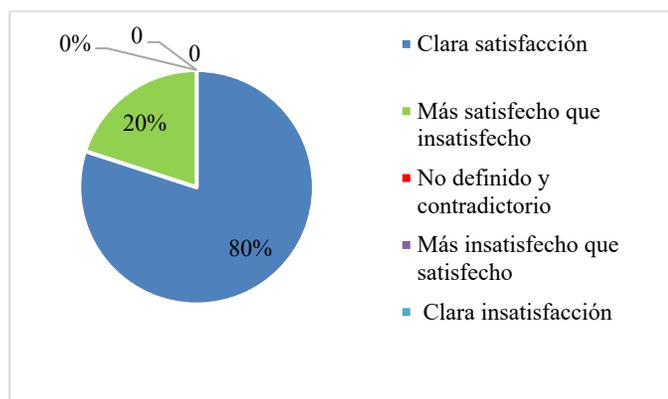


Figura 11: Valoración de usuarios sobre el proceso.

## 4. Conclusiones

- El análisis realizado a la literatura sobre los modelos, estándares y procesos de mejora en el área de pruebas

de *software* permitieron identificar buenas prácticas para definir los requisitos del proceso base de Pruebas de *software* y separarlos por niveles de madurez para su adopción de manera paulatina.

- La descripción gráfica y textual de los subprocesos de pruebas de *software* permite especificar las tareas que forman cada actividad, además de ejemplificarlas para un mayor entendimiento.
- La validación de la propuesta demostró que al implementar el proceso base de Pruebas de *software*, se aumentó la detención de defectos en etapas tempranas, por lo que disminuyeron las NC en el entorno operacional.
- Se recomienda analizar para futuras versiones del MCDAI una propuesta de niveles de implementación dentro de un mismo requisito específico, para que les permita a los proyectos optar por un nivel de forma objetiva y acorde a sus necesidades definidas.

## REFERENCIAS

- [1] R. S. Pressman. *Ingeniería del software. Un enfoque práctico*, Séptima edición ed., The McGraw-Hill Companies, 2010.
- [2] F. J. P.-C. J. M. M. Martha Lucia Rojas-Montes, "Proceso de pruebas para pequeñas organizaciones desarrolladoras de software," *Revista Facultad de Ingeniería (Fac. Ing.)*, vol. 24, no. 39, pp. 55-70, Mayo-Agosto 2015.
- [3] M. G. Velthuis, F. O. G. Rubio, I. G. R. d. Guzmán and F. Pino, *Calidad de Sistemas de Información*, 3ra edición ampliada y actualizada ed., Madrid: RA-MA, 2015, p. 217.
- [4] I. I. O. B. P. I. I. L. G. F. P. I. I. M. C. H. I. I. L. G. C. V. Ing. Roberto Carlos Castilla Blanco, "Proceso de pruebas y suite de herramientas de soluciones informáticas para la salud," *Revista Cubana de Informática Médica*, vol. 7, no. 1, pp. 56-72, 2015.
- [5] I. C. Society, *IEEE 1012-2004 Standard for Software Verification and Validation*, 2004.
- [6] M. Á. F. Fernández, *Aplicación de técnicas de pruebas automáticas basadas en propiedades a los diferentes niveles de prueba del software*, 2015.
- [7] D. S. J. Mayorga, *Análisis de métodos, técnicas y herramientas de verificación y validación de software aplicados a la Dirección de tecnología y comunicación de la Universidad técnica de Ambato, Ecuador*, 2017.
- [8] ISO/IEC/IEEE, *ISO/IEC/IEEE 12207 Ingeniería de Sistemas y Software— Procesos del ciclo de vida del Software*, 2017.
- [9] I. Sommerville, *Ingeniería de Software*, Madrid. España: PEARSON Education, 2011.

- [10] ISO/IEC/IEEE, *ISO/IEC/IEEE 29119-2 Software and systems engineering —Software testing —Part 2:Test processes*, Switzerland: ISO/IEC/IEEE, 2013.
- [11] G. R. S. Rex Black, *Fundamentos de pruebas de software*, Editorial RBCS, 2011.
- [12] R. Pressman, *Ingeniería de Software: Un enfoque práctico*, Madrid: McGraw Hill, 2002.
- [13] I. C. S. P. P. Committee, *SWEBOK, Guide to the Software Engineering Body of Knowledge*, United States of America, 2004.
- [14] M. P. V. C. M. F. S. Francisco J.Pino Correa, *Modelo de madurez de ingeniería de software*, España: AENOR, 2014.
- [15] ISQI, *Probador Certificado Nivel básico*, Berlin , 2016.
- [16] M. d. Justicia, "*Gaceta Oficial de la República de Cuba*," Consejo de Estado, La Habana. Cuba, 2019.
- [17] M. d. I. Comunicaciones, *Resolución 124 Reglamento para la producción de los programas y las aplicaciones informáticas y su evaluación de su calidad*, La Habana, 2019.
- [18] I. D. P. Montalván, *GUÍA GENERAL PARA UN MODELO CUBANO DE DESARROLLO DE APLICACIONES INFORMÁTICAS*, Universidad de las Ciencias Informáticas , 2014.
- [19] A. Febles, *Un modelo de Referencia para la Gestión de Configuración en la de Software*, Instituto Superior Politécnico "José Antonio Echeverría". Facultad de Ingeniería Industrial.Centro de Estudios de Ingeniería y Sistemas..
- [20] CALISOFT, *CS-03-D (14-001) LIBRO DE DIAGNÓSTICO*, La Habana , 2014.
- [21] Y. L. A. ., J. F. R. P. Leanet Tamayo Oro, "Proceso de diseño del software para un modelo de la calidad en Cuba," *Revista I+D Tecnológico*, vol. 15, no. 1, pp. 87-98, 2019.
- [22] CALISOFT, *CS-03-D (17-001) Libro de diagnóstico*, La Habana, 2017.
- [23] R. ACADEMIA, "Española, D. L. L. E Diccionario de la lengua española| Real Academia," 2014.
- [24] S. E. I. SEI, *CMMI para el desarrollo v1.3*, Cornege Mellon University, 2010.
- [25] H. Oktaba, M. Piattini, F. J.Pino, M. J. Orozco and C. Alquicira, *COMPETISOFT "Mejora de Procesos de Software para Pequeñas y Medianas Empresas y Proyectos"*, Madrid: Ra-Ma, 2008.
- [26] H. OKTABA and C. A. e. a. ESQUIVEL, *Modelo de Procesos para la Industria de Software MOPROSOFT*, México: Secretaría de Economía, 2005.
- [27] ISO/IEC/IEEE, *ISO/IEC/IEEE 29119-1 Software and systems engineering — Software testing — Part 1: Concepts and definitions*, 2013.
- [28] T. Foundation, *Test Maturity Model integration(TMMi)*, Irlanda: TMMi Foundation, 2015.
- [29] J. A. Mera-Paz, "*Análisis del proceso de pruebas de calidad de software*," *Ingeniería Solidaria*, vol. vol. 12, no. no. 20, pp. pp. 163-176, 2016.
- [30] D. G.-S. ., M. D. D. D. Dalila Jústiz-Núñez, "Proceso de pruebas para productos de software en un laboratorio de calidad," *Ingeniería Industrial*, vol. Vol. XXXV, no. No. 2, 2014.
- [31] P. Itti Hooda and P. Rajender Singh Chhillar, "Software Test Process, Testing Types and Techniques," *International Journal of Computer Applications (0975 – 8887)*, vol. 111, no. 13, 2015.
- [32] N. R. M. Reyes, "Reseña metodológica sobre los grupos focales," *Editorial Universidad Don Bosco*, , vol. 9, no. 6, pp. pp.47-53, 2012.
- [33] M. Aigner, "La técnica de recolección de información mediante los grupos focales," *Revista electrónica*, vol. 7, 2006.
- [34] I. Y. L. Alvarado, *Proceso Base de Aseguramiento de la Calidad para el Desarrollo de Software en Cuba*, Universidad de las Ciencias Informáticas, 2016.
- [35] S. G. Pérez, *Proceso de evaluación de la funcionalidad a productos de software en un laboratorio de pruebas*, 2017.
- [36] I. Y. S. Osorio, *PROCESO GESTIÓN DE ADQUISICIONES PARA LA INDUSTRIA CUBANA DEL SOFTWARE*, La Habana, 2017.
- [37] B. R. a. T. P. Katarína Hrabovská, "Software Testing Process Models Benefits & Drawbacks : a Systematic Literature Review," *arxiv.org*, 2019.