



Uso del framework de ISTQB para aplicaciones desarrolladas en una entidad bancaria: Caso de uso “Portabilidad de Nómina”

Use of ISTQB framework for applications development in a financial entity: Use case “Payroll Portability”

Francisco Nahúm Sánchez Ramírez

Universidad Politécnica de Tulancingo, Tulancingo de Bravo, Hidalgo, México
nahum.sanchez2231032@upt.edu.mx
ORCID: 0009-0008-0310-176X

José Manuel Hernández Reyes

Universidad Politécnica de Tulancingo, Tulancingo de Bravo, Hidalgo, México
jose.hernandez@upt.edu.mx
ORCID: 0000-0001-8642-2930

Carlos Enríquez Ramírez

Universidad Politécnica de Tulancingo, Tulancingo de Bravo, Hidalgo, México
carlos.enriquez@upt.edu.mx
ORCID: 0000-0003-4963-9828

doi: <https://doi.org/10.36825/RITI.12.27.003>

Recibido: Junio 22, 2024

Aceptado: Agosto 17, 2024

Resumen: En este trabajo se hace referencia al control de la calidad por medio de las pruebas de software, mediante la gestión de los elementos fundamentales que involucran a la *International Software Testing Qualifications Board* (ISTQB), utilizando la metodología de *Extreme Programming* (XP). Con la finalidad de implementar una estrategia de plan de pruebas para la verificación y validación de los procesos que se están llevando a cabo para la Portabilidad de Nómina en la Institución Financiera, lo cual permite la identificación de artefactos y/o documentos que apoyen la toma de decisiones para la alta dirección obteniendo como producto en el proceso de control de pruebas tanto individuales como de equipo de desarrollo.

Palabras clave: *ISTQB, Portabilidad de Nómina, Pruebas de Software, Extreme Programming, Metodología.*

Abstract: This paper refers to quality control through software testing, by managing the fundamental elements that involve the International Software Testing Qualifications Board (ISTQB), using the Extreme Programming (XP) methodology. The purpose of this paper is to implement a test plan strategy for the verification and validation of the processes that are being carried out for Payroll Portability in the Financial Institution, which allows the identification of artifacts and/or documents that support decision-making for senior management, obtaining as a product in the test control process both individually and as a development team.

Keywords: *ISTQB, Salary Portability, Software Testing, Extreme Programming, Methodology.*

1. Introducción

El objetivo principal de un plan de pruebas es orientar el esfuerzo del control de calidad de un producto de software a través de la identificación de los procesos detallados de acuerdo con la relevancia de los requerimientos, enfocando la ejecución de las pruebas a la funcionalidad del producto de software [1].

Las pruebas de software incorporan los principios y la gestión de ISTQB, las cuales consisten en prácticas utilizadas para mejorar la calidad de las pruebas unitarias [2]; incorporando herramientas auxiliares al *testing* optimizando la calidad del código y permitiendo mitigar rápidamente los defectos. La metodología *XP* aplicada, se basa en los elementos que componen el producto así como los procedimientos de pruebas de casos similares con buenos resultados. Además, de la incorporación de herramientas de automatización en la metodología durante el ciclo de vida del software [3].

El uso de herramientas de apoyo, mejor conocidas como de automatización, ayuda a la mejora de tiempos de entrega y diseño de procesos, para agilizar los casos de prueba y generar documentación, que brinden un sustento en la toma de decisiones de forma rápida y efectiva, siendo de gran apoyo para casos imprevistos [4]. Se incorporan apartados del trabajo actual y elementos de importancia, cual es la mejora y el propósito que se tiene con estas incorporaciones a los casos actuales.

2. Estado del arte

Existe diversas formas de interpretar los elementos y componentes que se asemejan al proceso que se desea cubrir, para ello es necesario tener el panorama específico de las fortalezas a lograr en la calidad de los productos de software.

De acuerdo con el avance de la tecnología, son diversas las necesidades que los procesos de software pueden cumplir, sin embargo, la garantía de que realmente se cumpla con los requisitos y con la calidad establecida en el alcance, así como los objetivos del plan de pruebas difiere de las necesidades, para ello, es necesario realizar la implementación de las prácticas que aseguren la calidad establecida en el producto.

En los productos financieros existen muy pocas métricas descritas, aun así, es posible identificar algunos autores que asemejan esta medición como a continuación se describe: Jiménez Vargas Pedro Augusto y Fartolino Guerrero Freddy en su tesis “Solución de automatización de pruebas de calidad en empresa bancaria para reducir tiempos de certificación con herramientas tecnológicas de software”, se enfrenta desafíos en el proceso de pruebas de calidad, lo que retrasa la entrega de productos y servicios [5]. Propone una solución que involucra *Selenium*, suites de pruebas y pruebas automatizadas en un entorno de integración continua. La arquitectura física aborda la infraestructura de hardware y entornos de prueba, garantizando la confiabilidad y eficiencia, implementó su metodología y obtuvo los siguientes resultados:

- La problemática, Dificultad para mantener un seguimiento claro y continuo entre los requisitos y las pruebas realizadas.
- Personal de interés, Gerente TI, Subgerente TI, Ingeniero de calidad, Ingeniero de sistemas, Analista de certificación y analista de automatización.
- Que fue lo que se aplicó, Automatización de pruebas funcionales y de regresión, se generaron casos de prueba, reportes y resultados, así como la integración continua.
- Los resultados obtenidos, aplicación del coeficiente alfa de Cronbach, una métrica reconocida en la evaluación de la consistencia interna y fiabilidad de los datos. Cada experto calificó los aspectos en una escala de 1 a 5, expresando su opinión sobre el desempeño o calidad en cada evaluación específica.

Un segundo caso es en el que se implementan las prácticas ISTQB, como lo menciona Roberth Melgar Velázquez en su tesis que lleva por título “Implementación de un modelo de gestión de pruebas de software según ISTQB para mejorar el proceso del área de certificación en tecnologías web de una entidad financiera” describe que las entidades financieras tienen como foco principal cubrir las necesidades del cliente, además, indica que en el sector financiero se ha incrementado la presión competitiva [6].

En su implementación, obtuvo los siguientes resultados:

- Problemática, se definieron herramientas para la gestión de pruebas de software y así mejorar el control de la atención de los requerimientos, se establecieron fases de pruebas, se aplicaron prácticas ISTQB para aumentar la fiabilidad de las pruebas.

- Que fue lo que se aplicó, Pruebas de certificación, pruebas de funcionalidad, pruebas de ratificación, pruebas de reversión.
- Los resultados obtenidos, se mejoró el control y la transparencia en los procedimientos, al aplicar ISTQB se comprende mejor el cambio en los requerimientos, además, la estructura base para los casos de prueba proporcionan facilidad en el diseño de escenarios de casos.

Actualmente, existen diferentes tipos de herramientas que pueden ayudar a automatizar tareas de pruebas de software. La mayoría de las herramientas son de costo, aunque también hay algunas libres, sin embargo, es necesario tener en cuenta que se deben definir entradas y se debe conocer las salidas esperadas. Además, es importante establecer las causas donde aplicar los casos de prueba, debido a que no es posible la automatización de todos los escenarios probables.

Para dar uso a las herramientas es necesario tener un conocimiento previo, de esta manera es posible obtener un mayor provecho posible al momento de implementarlas, así como lo menciona Elian Andrés Díaz Vargas en su tesis “Diseño de un sistema de pruebas para evaluación de funcionalidad y desempeño del software desarrollado por la empresa TCS *Solution Center*” menciona que se requiere realizar pruebas de software para comprobar que las funcionalidades de las aplicaciones desarrolladas por la empresa se ejecutan correctamente, son acordes con los requerimientos dados por el cliente y no tienen fallos importantes que puedan ocasionar que funcione diferente a como se diseñó o que luego de que la aplicación salga a producción se encuentren problemas inesperados que hagan que la aplicación falle [7].

Durante la incorporación del modelo obtuvo los siguientes resultados:

- La problemática, realizar pruebas de software para comprobar las funcionalidades de las aplicaciones desarrolladas por la empresa, acorde a los requerimientos dados por el cliente y no tienen fallos importantes que puedan ocasionar que funcione diferente a como se diseñó para cumplir con la calidad esperada por el cliente.
- Que fue lo que se aplicó, la metodología BDD (*Behavior Driven Development*) para mejorar la comunicación entre todas las partes involucradas y se centra en el comportamiento del sistema, pruebas unitarias, de regresión, integración, UI, performance, pruebas de seguridad, pruebas E2E (de acuerdo a las reglas de negocio), pruebas exploratorias, patrón de diseño *Screenplay*.
- Los resultados obtenidos, paquetes *questions, taks, user interfaces, utils, features, runners, step definitions* estandarizados para *screenplay*, informes realizados por *Serenity* generados de manera automática.

De acuerdo con lo descrito por los autores y considerando la practicidad de herramientas auxiliares para facilitar el proceso de pruebas de software, debemos considerar todos los factores que puedan facilitar los procesos de prueba, además, establecer inicialmente los alcances y las limitaciones.

3. Materiales y métodos

En esta sección se describe los aspectos relacionados con el plan de pruebas que se aplican en el caso de portabilidad de nómina.

3.1. Principios de International Software Testing Qualifications Board

La aplicación de los siete principios de ISTQB está generando resultados, se han encontrado varios casos que en la descripción de los principios son muy puntuales y de gran ayuda cuando se realizan en cada fase del ciclo de vida, estos son [8]:

- En las pruebas unitarias se han logrado identificar defectos, también se ha logrado mejorar el proceso de desarrollo ejecutando mejores prácticas con herramientas de automatización.
- No se están trabajando pruebas exhaustivas, el proceso unitario ayuda a que la parte medular permita identificar estos defectos y corregirlos en fases tempranas, esto nos ayuda a mejorar los tiempos ya que se corrigen los defectos que se van encontrando, esto de acuerdo con la metodología aplicada, XP, ya que lo permite.
- Se trata también de minimizar la paradoja de pesticida, por ello es que se solicitan estratos de datos que permitan evaluar los casos con distintos datos, esto nos dará mejores resultados ya que permitirá obtener nuevos defectos con información distinta.

- Las pruebas realizadas corresponden al desarrollo, con ello se permite la corrección de los defectos localizados al momento.
- La actualización de los procesos de prueba y llevarlos a cabo en fases tempranas permite identificar la mayor cantidad de defectos, sin embargo, debemos ser conscientes de que es posible que surjan contratiempos en algunos casos.

3.2. Herramientas de automatización

En la actualidad existen herramientas y programas que ayudan a facilitar las tareas de pruebas de software, cabe mencionar que no todas pueden ser automatizadas ya que muchas veces es necesario que sean valoradas por el cliente o el equipo en cuestión; el uso de las herramientas de apoyo e implementación en los casos de prueba es necesario considerar dos escenarios que ayudarán a determinar el costo beneficio para la organización [9]. En el primer escenario se encuentran las herramientas que son comerciales, para poder hacer uso de ellas es necesario adquirir la herramienta o extensión a través de un pago. También se encuentran las herramientas de uso libre, la mayoría son de desarrollo comunitario y son complementadas por desarrolladores de distintos países.

Las extensiones y la aplicación de herramientas de apoyo en la elaboración de pruebas se tomaron en cuenta los entornos de desarrollo, así como los requisitos del proyecto, se verificó si se tiene la extensión o soporte para cada entorno, además de examinar cada una de ellas documentalmente y así poder determinar las necesarias. Realizado el proceso descrito, se determinó el uso de las siguientes herramientas y extensiones [10]:

- *NUnit Test*
- *Snyk Security*
- *SonarLint*
- *VSSonarExtension*
- *Unit Test Boilerplate Generator*
- *Owasp Zap*
- *Jaeger*
- *Mantis Bug Tracker*

3.3. Antecedentes de la calidad en la entidad financiera

Actualmente se cuenta con algunos datos que ayudan a obtener un conocimiento de las pruebas elaboradas, sin embargo, es necesario complementarlas ya que institucionalmente son adecuados, pero existen procesos que deben ser gestionados para atender solicitudes de otros organismos financieros que deberán cumplir con sus normas, lo importante es tener establecidos los procesos de prueba para tener un punto de apoyo en caso de surgir inconvenientes, motivo por el cual la implementación de XP es muy conveniente, debido a que provee de integración continua para cada apartado a probar y esto garantiza que las correcciones sean inmediatas y aumenta la calidad del proceso de entrega.

Se tiene como documento principal la atención de las incidencias, se encuentran registrados en un documento de Excel, en el documento se registran elementos como:

1. Id del hallazgo
2. Módulo
3. Fecha
4. Quien realiza la prueba
5. Requerimiento
6. Descripción de la incidencia
7. Evidencia (hipervínculo)
8. Severidad
9. Estatus
10. Comentarios
11. Quien debe resolver la incidencia

En donde el apartado de evidencia relaciona hojas que se van generando conforme se va realizando el proceso de calidad, también se permite la incorporar imágenes como de la prueba realizada, se muestra parte de los datos en la Figura 1.

Caso	Incidencia	Hipervínculo (EVIDENCIA)	Severidad	Acepta (Se Escala)	Rechaza (Causa)	Estatus Pruebas
	Formulario de cambio de contraseña: Usuario muestra que el usuario es el correo y no es el usuario ocupado para loguearte	Incidencia 1	Alta			Cerrada
	Te envía un Modal de código erróneo "El código de verificación no coincide con el correo. Vuelva a intentarlo" y te regresa a volver a escribir tu correo	Incidencia 2	Alta			Cerrada

Figura 1. Registro de hallazgos de las pruebas en la actualidad.

Una vez que se corrigen los hallazgos de las pruebas y estas son satisfactorias, el departamento que realizó la solicitud del módulo o proyecto suele llevar a cabo una segunda prueba (repetición de la prueba), en un ambiente pre productivo, si se encuentran nuevos elementos que deban ser corregidos se llenan nuevamente los documentos junto con las evidencias, en caso de que no existan más hallazgos se da la firma aceptación de conformidad que consiste en un documento de Pruebas de Aceptación del Usuario (*User Acceptance Testing* - UAT por sus siglas en Inglés) [11], este documento contiene:

- Información general del producto o proyecto
- Certificado del producto, consiste en la descripción de los documentos generados, elementos a los que se les asegura la calidad, evidencias generales de las pruebas realizadas, las firmas de aprobación y aceptación del proyecto, módulo o producto, en la Figura 2 Documento UAT.

I. Información General del Producto:

ID Orden de Trabajo:	OT-0000-202X
Identificador del Proyecto:	OT-0000-202X Nombre del Proyecto
Nombre del Proyecto:	Nombre del proyecto
Líder Funcional:	Nombre del encargado
Puesto del Líder Funcional:	Puesto del encargado
Responsable de A. de la Calidad:	Nombre del responsable de calidad
Puesto Responsable de A. de la Calidad	Puesto del responsable de calidad

Figura 2. Información general UAT.

Esta es la manera en la que actualmente se garantiza la calidad, como mención importante a considerar, las aplicaciones han estado evaluadas únicamente para uso interno, motivo por el cual es necesario una mejora para la medición de la calidad del software desarrollado, motivo por el cual la incorporación de una metodología, así como ISTQB y las herramientas de automatización que ayudarán a mejorar la evaluación del proceso de calidad de los productos [2, 3, 10].

3.4. Metodología aplicada

La implementación de XP inicialmente permitió que durante la generación de la planeación se identificaran responsabilidades que debía tomar el equipo y otras que son responsabilidad del cliente, en donde el valor del proyecto y su información es determinado por el cliente, y el equipo es el encargado de determinar los datos de costos y beneficios [2].

Determinadas las responsabilidades y aplicado a elementos que incorpora XP se obtuvieron las siguientes mejoras en la entidad financiera, aplicado al caso de uso “Portabilidad de Nómina”.

1. Retroalimentación rápida, durante cada caso de pruebas y una vez que se obtuvieron los resultados se realizó la solicitud de una retroalimentación por el cliente, en caso de haber alguna mejora o existir detalles en el proceso se atendió de manera inmediata como lo determina el proceso metodológico en cada fase.
2. Simplicidad en la atención, cada caso de prueba se atendió de acuerdo con lo planeado, ya que existen procesos prioritarios y que requieren mayor atención de acuerdo con el nivel de complejidad y usabilidad.
3. Cambios incrementales, la atención de defectos en las pruebas unitaria genera una simplicidad única, ya que es mejor atender defectos en cada proceso unitario que generarlo y atenderlo en un módulo completo, debido a que en él se incorporan varios procesos unitarios, esto genera una lista de métodos que deben ser atendidos en caso de encontrar un defecto al probar el módulo completo.
4. Adopción del cambio, en caso de tener requisitos nuevos proporcionados por el cliente, estos se atienden y se planifica la implementación, caso ocurrido en la generación de la “Portabilidad de Nómina” una segunda aplicación.
5. Planificación, mejora la comunicación del equipo ya que en caso de depender de algún proceso elaborado por otro miembro se deberá trabajar en conjunto para así ver integralmente el comportamiento.
6. Liberaciones ligeras, periódicamente el equipo realiza liberaciones frecuentes con las mejoras realizadas y los defectos atendidos.
7. Explicación mediante metáforas, en nuevas incorporaciones al equipo de desarrollo, se realiza una explicación de diseño en vez de atender lecturas ya que esto agiliza el proceso de capacitación, además de que se proporcionan datos elementales para las labores.
8. Desarrollo guiado por pruebas, se incorpora el TDD (*Test Driven development* - Desarrollo Guiado por Pruebas) lo que permite mejorar la calidad del código a través de pruebas.
9. Refactorización, para mejorar la calidad del código sin alterar la funcionalidad.
10. Programación en pares, en algunos casos el trabajo en parejas es adecuado ya que permite tener más ideas del mejoramiento del comportamiento y así optimizar el rendimiento en los procesos.
11. Historias de usuario, para una mayor simplicidad de las necesidades.
12. Implantación de estándares de codificación, que son pilares en los procesos de desarrollo de software en la organización.

Se establece un proceso de mejora para la fase de prueba de cada liberación del producto estandarizado en la organización, el cual es representado en la Figura 3.

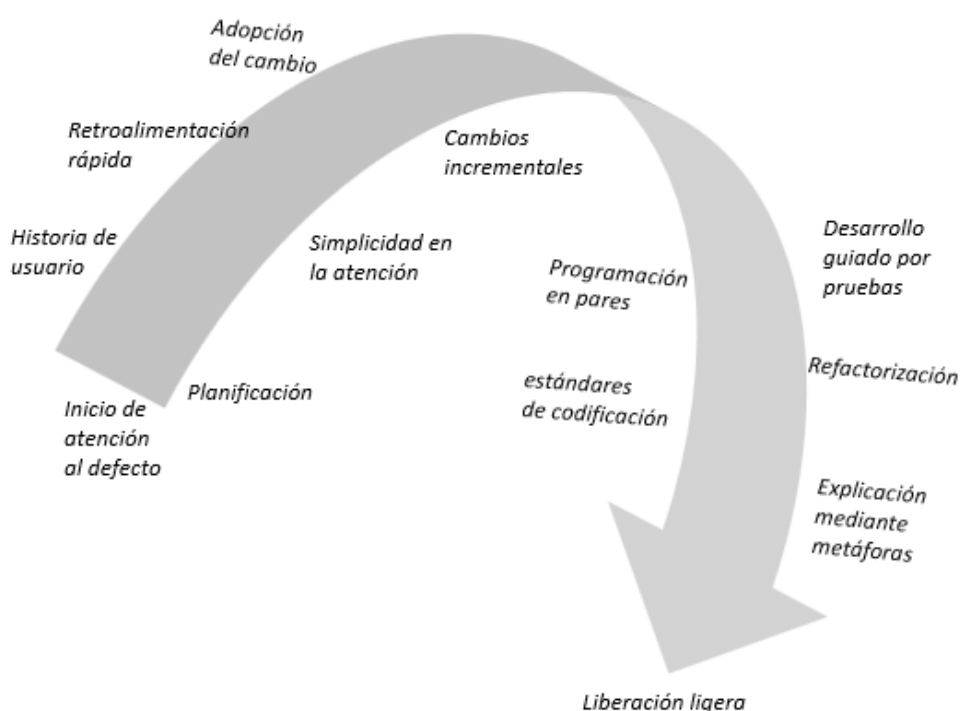


Figura 3. Proceso de mejora en la fase de prueba de cada liberación del producto.

4. Resultados

Con la incorporación del modelo XP se complementa lo que ya existe en la institución financiera que, hasta ahora, ha funcionado de manera satisfactoria. La incorporación del cliente en algunas de las pruebas es importante, la prioridad para generar las pruebas de los procesos es el punto inicial de ello, ya que de acuerdo a esto conoceremos que tareas tienen mayor importancia y exigencia, esto permitirá elaborar una mejor planeación para el aseguramiento de la calidad en tareas específicas y que requieren una mayor atención.

Para el control y aseguramiento de la calidad se incorporan los siguientes tipos de pruebas a lo que ya existe en la institución financiera:

1. Pruebas unitarias, ayuda al aseguramiento del proceso desde la parte más básica que es la función [12].
2. Pruebas integrales, para asegurar que cada bloque unitario es capaz de realizar el proceso en conjunto con otras acciones para llevar a cabo una tarea conjunta [13].
3. Pruebas robustas, permitirá medir el comportamiento en la obtención de los resultados [14].
4. Pruebas de estrés, para asegurar que no existan inconsistencias en el uso del producto, módulo o aplicación, además de probar los límites del sistema [15].
5. Pruebas de funcionalidad, valida que el proceso, vista, envío y recepción de datos sea correcto y acorde a lo que el cliente necesita [16].

Las pruebas unitarias realizadas considerando el proceso de gestión de ISTQB han generado los siguientes resultados:

- Se ha realizado una planificación, que consiste en definir las pruebas unitarias de mayor relevancia de acuerdo con la importancia del módulo para el cliente.
- Para llevar a cabo el monitoreo y el control de pruebas se generan versiones de las aplicaciones donde los históricos permiten ver los defectos corregidos y la afectación que produjo.
- El análisis de las pruebas de unidad, se determinan de acuerdo con el nivel de importancia que el cliente considera (alto, medio, bajo).
- Cada prueba es diferente, cada elemento unitario tiene una descripción específica y permite identificar que ocurrió durante la misma.
- Para su ejecución se trabaja con segmentos de datos ya que no es posible tener todos los elementos, esto por la confidencialidad del cliente.

Las herramientas que se utilizan para el aseguramiento de la calidad:

- Extensiones
 - NUnit Test, *framework* para la elaboración y ejecución de pruebas unitarias [17].
 - Snyk Security, contribuye a la búsqueda de problemas de seguridad en el código [18].
 - SonarLint, detectar y soluciona problemas de calidad en el código mientras se escribe [19].
 - VSSonarExtension, para realizar un análisis de acuerdo con las reglas estáticas establecidas y configuradas [19].
 - Unit Test Boilerplate Generator, permite la generación de pruebas unitarias para .NET [20].
- Verificación de vulnerabilidades
 - Owasp Zap, escáner de seguridad web. Es utilizado como una aplicación de seguridad para pruebas de penetración [21].
- Aseguramiento de servicios
 - Jaeger, monitoreo y solución de problemas en la comunicación de microservicios [22].
- Aseguramiento de calidad en el código
 - Mantis Bug Tracker, detección de errores durante el ciclo de vida del defecto [23].

Se incorporaron nuevas tareas de prueba y herramientas de apoyo que permite tener un mayor control de calidad en cada aspecto de la arquitectura de las aplicaciones, la incorporación de contenedores admite la distribuir de las tareas en procesos. Aunado a las entradas y salidas que proporciona cada herramienta, se incorporaron documentos, en ellos se define una estructura que contiene los resultados de las pruebas realizadas, a fin de tener evidencias de las métricas de calidad aplicadas en los términos establecidos a lo largo del ciclo de vida del proyecto.

El caso de uso “Portabilidad de Nómina”, para la transferencia de prestaciones laborales, de la Institución Financiera, se encuentra en una fase de pruebas unitarias iterativas, a ellas se incorporan los principios señalados por ISTQB, el entorno de desarrollo es un NET Core 8 para las aplicaciones, incorporando servicios WSO2, un NGINX para recepción y evaluación de peticiones, además de un ambiente controlado por Docker (ambiente de aplicaciones a través de contenedores), lo cual agiliza los procesos de prueba al estar seccionados es posible evaluar cada parte de manera independiente [24, 25, 26, 27]. Los resultados:

- Detección de código que puede ser optimizado.
- Reducción de sentencias.
- Mejora en el rendimiento y agilización de peticiones en cada segmento de código.
- Detección de errores muy puntuales y una muy rápida solución.
- Integración más sencilla.
- Reducción de errores en otros segmentos de código.
- Nuevas aportaciones sin alterar una integración.
- Una más rápida y fácil detección de errores.

Como medidas de calidad de acuerdo con lo planeado para la generación de los productos de software a desarrollar, estas deberán ser expuestas y evaluadas por organismos que incorporan normas de calidad, se muestran resultados de datos obtenidos en las versiones de desarrollo de defectos registrados en algunos módulos en la Tabla 1.

Tabla 1. Resultados preliminares del proceso de iteración de pruebas propuesto.

Módulo	Defectos en desarrollo	Defectos en pruebas unitarias	Mejoras en el código	Defectos en prueba de regresión
Consultas	10	10	8	0
Solicitudes	5	10	5	0
Seguimiento	8	7	8	0
Cancelación	15	15	7	0

Fuente: Datos obtenidos del versionado de la aplicación.

Los datos mostrados en la tabla anterior están representados de la siguiente manera:

- *Módulo*, estructura de atención para llevar a cabo una tarea o conjunto de tareas.
- *Defectos en desarrollo*, atención de validaciones y defectos encontrados durante la creación de una tarea unitaria o conjunto de tareas.
- *Defectos en pruebas unitarias*, obtención de los defectos encontrados al realizar pruebas de unidad.
- Mejoras en el código, para ello se utilizan herramientas que ofrecen recomendaciones de mejora y es responsabilidad del desarrollador tomarlas o descartarlas.
- *Defectos en pruebas de regresión*, repetir la prueba para visualizar que el defecto se atendió satisfactoriamente.

Una vez obtenido los defectos de las pruebas de regresión en ceros, es óptimo llevar a cabo las pruebas continuas cada que se incorpore un incremento del producto.

5. Conclusiones

En el presente artículo se explora las pruebas de software, así como la interpretación y contextualización de la importancia del *testing*, en las aplicaciones o productos finales en atención a requerimientos de clientes. Se propone la incorporación de un escenario en donde se involucran prácticas para el aseguramiento de la calidad a través de pruebas implementadas en el entorno financiero, esto permite obtener mayor calidad en las aplicaciones, mejorar el aspecto documental, al incorporar herramientas de automatización, se agilizarán algunos casos de prueba, además, las evidencias documentales son más completas y se verán reflejados en las evidencias que se obtienen. El material generado en este artículo representa el avance de una investigación, generación de un escenario aplicado en proyectos con enfoques financieros, además, se pretende que no se aplique en una única ocasión, ya que el proceso financiero es enorme y requiere de atenciones de acuerdo con el avance tecnológico.

La metodología es adaptable, de acuerdo con los escenarios que se presenten ya que cada desarrollo difiere en muchos elementos, lo que significa que también es repetitivo y suele ser funcional con cambios mínimos sin alterar la parte medular metodológica utilizada.

La atención de trabajos futuros se plantea de la siguiente manera:

- Seguimiento de la metodología en el proceso de pruebas para el caso de uso “Portabilidad de Nómina”, misma que está regida por normas establecidas por BANXICO, que es el organismo encargado de realizar las gestiones de las instituciones bancarias en México.
- Generación de recursos documentales que proporcione el seguimiento de los procesos y evidencias de cada uno de los resultados.
- Implementación del proceso de pruebas en proyectos futuros, no solo con un enfoque financiero, el objetivo es consolidar la propuesta y que sea aplicable a distintos proyectos y ayude a mejorar la atención de pruebas en reducción de tiempos y apoyo a los departamentos de desarrollo y QA.

Para que el proceso de pruebas en conjunto con la metodología tenga los resultados más certeros, es importante tener en cuenta que el análisis de cada elemento que se ve a involucrar esté bien orientado y delimitado, como es del conocimiento del personal de desarrollo y pruebas, no es posible probar todo, es por eso que es necesario tener bien específicas las reglas y así poder determinar que el plan de riesgos cumplirá con procesos que posiblemente puedan causar inconsistencias.

6. Referencias

- [1] Guardati, S., Ponce, A. (2011). Guía de pruebas de software para MoProSoft. REICIS. *Revista Española de Innovación, Calidad e Ingeniería del Software*, 7 (2), 28-47.
- [2] Andreas Spillner, T. I. (2021). *Software Testing Foundations* (5th Ed.). Stefanie Weidner.
- [3] Velásquez, S. M., Vahos Montoya, J. D., Gómez Adasme, M. E., Restrepo Zapata, E. J., Pino, A. A., Londoño Marín, S. (2019). Una revisión comparativa de la literatura acerca de metodologías tradicionales y modernas de desarrollo de software. *Revista CINTEX*, 24 (2), 13-23. <https://doi.org/10.33131/24222208.334>
- [4] Serna, E., Martínez, R., Tamayo, P. (2019). Una revisión a la realidad de la automatización de las pruebas del software. *Computación y Sistemas*, 23 (1), 169-183. <https://doi.org/10.13053/cys-23-1-2782>
- [5] Jimenez Vargas, P. A., Fartolino Guerrero, F. (2023). *Solución de automatización de pruebas de calidad en empresa bancaria para reducir tiempos de certificación con herramientas tecnológicas de software* [Tesis de Grado]. Universidad Peruana de Ciencias Aplicadas (UPC), Lima. <http://hdl.handle.net/10757/670906>
- [6] Melgar Velásquez, R. (2020). *Implementación de un modelo de gestión de pruebas de software según ISTQB para mejorar el proceso del área de certificación en tecnologías web de una entidad financiera* [Tesis de Grado]. Universidad Tecnológica del Perú, Lima. <https://hdl.handle.net/20.500.12867/3290>
- [7] Díaz Vargas, E. A. (2023). *Diseño de un sistema de pruebas para evaluación de funcionalidad y desempeño del software desarrollado por la empresa TCS Solution Center* [Tesis de Doctorado]. Universidad Santo Tomás, Bucaramanga, Colombia. <http://hdl.handle.net/11634/50548>
- [8] Kramer, A., Legeard, B. (2016). *Model-Based Testing Essentials: Guide to the ISTQB Certified Model-Based Tester Foundation Level*. John Wiley & Sons.
- [9] García Ruiz, J. (2017). *Estudio de herramientas de automatización de pruebas software en aplicaciones web* [Tesis de Grado]. Universidad de Valladolid, España. <http://uvadoc.uva.es/handle/10324/27484>
- [10] Dustin, E., Rashka, J., Paul, J. (1999). *Automated software testing: introduction, management, and performance*. Addison-Wesley Professional.
- [11] Cimperman, R. (2006). *Uat defined: A guide to practical user acceptance testing (digital short cut)*. Pearson Education.
- [12] Francisco Fernández, M. Á. (2015). *Aplicación de técnicas de pruebas automáticas basadas en propiedades a los diferentes niveles de prueba del software* [Tesis de Doctorado]. Universidad de la Coruña, España. <https://ruc.udc.es/dspace/handle/2183/14814>
- [13] Sommerville, I. (2005). *Ingeniería del software*. Pearson educación.
- [14] Lei, B., Li, X., Liu, Z., Morisset, C., Stolz, V. (2010). Robustness testing for software components. *Science of Computer Programming*, 75 (10), 879-897.

- [15] Sanchez Almenares, L. (2012). Prueba Automática de Carga y Estrés. *Serie Científica de la Universidad de las Ciencias Informáticas*, 5 (1), 1-8. <https://dialnet.unirioja.es/servlet/articulo?codigo=8590093>
- [16] Vargas, C. (2020). *Tipos de pruebas funcionales para el aseguramiento de la calidad*. <https://trycore.co/transformacion-digital/tipos-de-pruebas-funcionales/>
- [17] Pearman, G., Goodwill, J. (2006). Test Environment Tool: NUnit. En *Pro. NET 2.0 Extreme Programming* (pp. 59-67). Springer Link.
- [18] Tundis, A., Mazurczyk, W., Mühlhäuser, M. (2018). *A review of network vulnerabilities scanning tools: Types, capabilities and functioning*. 13th international conference on availability, reliability and security (pp. 1-10), Hamburg, Germany. <https://doi.org/10.1145/3230833.3233287>
- [19] Páez Marcote, M. (2023). *Plataforma para la gestión de la infraestructura orientada a proyectos* [Trabajo de Grado]. Universidad de la Coruña, España. <https://ruc.udc.es/dspace/handle/2183/33389>
- [20] Chowdhury, K. (2017). *Mastering Visual Studio 2017*. Packt Publishing Ltd.
- [21] García, R. E., Zenteno, A. C., Santiago, M. C., Romero, Y., Pérez, J., Rubín, G. T. Análisis de Herramientas para el Testeo de Vulnerabilidades de Aplicaciones Web. En G. T. Rubín Linares (Coord.) *Aportaciones de las Ciencias Computacionales Durante la Pandemia COVID 19* (pp. 46-51). Montiel & Soriano Editores S.A. de C.V. https://conacic.sycise.org/docs/LIBRO_CONACIC_2021_APORTACIONES.pdf#page=47
- [22] Shkuro, Y. (2019). *Mastering Distributed Tracing: Analyzing performance in microservices and complex systems*. Packt Publishing Ltd.
- [23] Kumar, R., Kumar Pani, P. (2020). A Comparative Analysis of Multiple Defect Tracking Systems. *Fuzzy Systems and Soft Computing*, 5 (1), <https://www.fuzzy-tversu.com/Vol.05,%20Issue.%202,%20July-December%20:%20%202020/105.pdf>
- [24] Krill, P. (2023). *Microsoft lanza.NET 8, la nueva versión de su plataforma de desarrollo de aplicaciones*. <https://link.gale.com/apps/doc/A773034263/IFME?u=anon~25a60db9&sid=googleScholar&xid=27815198>
- [25] García Represa, J. Larrinaga, F., Varga, P., Ochoa, W., Perez, A., Kozma, D., Delsing, J. (2023). Investigation of Microservice-Based Workflow Management Solutions for Industrial Automation. *Applied Science*, 13 (3), 1-28. <https://doi.org/10.3390/app13031835>
- [26] Fernández, C. A., Merlo, V. S., Jordan, M. A., José Rios, L. (2023). Análisis de balanceo de carga con Nginx. *Memorias de las JAIIO*, 9 (6), 208-217. <https://sedici.unlp.edu.ar/handle/10915/165890>
- [27] Fernandez Isern, G., Colomer Vila, J., Hernández Olasagarre, M. B, Blanco García, E. (2023). *Entornos y contenedores*. <https://eines-informatiques.recursos.uoc.edu/entornos-y-contenedores/es/2-1-introduccion/>