

**Documento con las verificaciones de condiciones de calidad del producto de software  
ajustado.**

GA11-220501098-AA3-EV02.

Carlos Arley López Ramírez

Centro de Servicios Financieros, SENA Regional Distrito Capital

2627072: Tecnólogo Análisis y Desarrollo de Software (ADSO)

Instructor: Andrés Rubiano Cucarían

14 de marzo de 2024

## Tabla de Contenido

<b>Índice de tablas.....</b>	<b>8</b>
<b>Documento con las verificaciones de condiciones de calidad del producto de software ajustado.....</b>	<b>9</b>
<b>1 Objetivo .....</b>	<b>10</b>
<b>1.1 Objetivos específicos:.....</b>	<b>10</b>
<b>2 Bitácora de los procesos documentales.....</b>	<b>11</b>
<b>3 Informe de Resultados del Comportamiento del Software - Easy Credit .....</b>	<b>13</b>
<b>3.1 Alcance de las Pruebas .....</b>	<b>13</b>
<b>3.2 Resumen Ejecutivo.....</b>	<b>13</b>
<b>3.3 Metodología .....</b>	<b>13</b>
3.3.1 <i>Equipo Evaluador .....</i>	<i>13</i>
EasyEval Team.....	13
3.3.2 <i>Recursos Utilizados.....</i>	<i>14</i>
3.3.2.1 <i>Herramientas de Pruebas: .....</i>	<i>14</i>
3.3.2.2 <i>Ambiente de Pruebas:.....</i>	<i>14</i>
<b>3.4 Ponderaciones.....</b>	<b>14</b>
3.4.1 <i>Ponderación de Métricas:.....</i>	<i>14</i>
3.4.2 <i>Ponderación de Pruebas:.....</i>	<i>14</i>
<b>3.5 Resumen de Pruebas.....</b>	<b>15</b>
3.5.1 <i>Pruebas de Funcionalidad:.....</i>	<i>15</i>
3.5.2 <i>Pruebas de Rendimiento: .....</i>	<i>15</i>
3.5.3 <i>Pruebas de Usabilidad:.....</i>	<i>15</i>
3.5.4 <i>Pruebas de Seguridad: .....</i>	<i>15</i>

<b>3.6</b>	<b>Resultados .....</b>	<b>16</b>
3.6.1	<i>Métricas de Rendimiento.....</i>	16
3.6.1.1	Tiempo de Carga de la Interfaz: .....	16
3.6.1.2	Tiempo de Respuesta de Transacciones: .....	17
3.6.2	<i>Métricas de Usabilidad .....</i>	19
3.6.2.1	Tiempo de Completar el Formulario de registro de clientes: .....	19
3.6.2.2	Tasa de Éxito en la Primera Interacción: .....	19
3.6.3	<i>Pruebas de Seguridad .....</i>	22
3.6.3.1	Número Total de Vulnerabilidades Detectadas: .....	22
3.6.3.2	Gravedad de las Vulnerabilidades: .....	22
3.6.3.3	Tipos de Vulnerabilidades: .....	22
3.6.4	<i>Métricas Adicionales.....</i>	24
3.6.4.1	Número de Transacciones Simultáneas: .....	24
<b>3.7</b>	<b>Hallazgos y Problemas.....</b>	<b>24</b>
3.7.1	<i>Rendimiento: .....</i>	24
3.7.2	<i>Usabilidad:.....</i>	24
3.7.3	<i>Funcionalidad: .....</i>	25
3.7.4	<i>Seguridad: .....</i>	25
<b>3.8</b>	<b>Comparación con Requisitos y Expectativas.....</b>	<b>25</b>
<b>3.9</b>	<b>Resumen de Defectos Detectados.....</b>	<b>25</b>
<b>4</b>	<b>Selección de buenas prácticas de calidad .....</b>	<b>26</b>
4.1	<b>Implementación de Pruebas Automatizadas:.....</b>	<b>26</b>
4.2	<b>Revisiones de Código Regular: .....</b>	<b>26</b>
4.3	<b>Seguimiento de Métricas de Calidad: .....</b>	<b>26</b>
4.4	<b>Retroalimentación de Usuarios: .....</b>	<b>27</b>

<b>5 Descripción de Características, Subcaracterísticas, Propiedades de Calidad y Métricas Utilizadas .....</b>	<b>27</b>
<b>5.1 Usabilidad:.....</b>	<b>27</b>
<b>5.2 Funcionalidad:.....</b>	<b>28</b>
<b>5.3 Rendimiento: .....</b>	<b>28</b>
<b>5.4 Seguridad:.....</b>	<b>28</b>
<b>6 Informe para realizar ajustes en los procesos de desarrollo .....</b>	<b>29</b>
<b>6.1 Alcance .....</b>	<b>29</b>
<b>6.2 Metodología seleccionada.....</b>	<b>29</b>
6.2.1 <i>Personal Software Process (PSP):.....</i>	29
<b>6.3 Cambios propuestos.....</b>	<b>30</b>
6.3.1 <i>Mejora de la Planificación e integración temprana de pruebas unitarias:.....</i>	30
6.3.2 <i>Optimización de Procesos:.....</i>	30
6.3.3 <i>Mejora de la Calidad del Código: .....</i>	31
6.3.4 <i>Promoción de la Colaboración y Comunicación:.....</i>	31
<b>6.4 Documentar los cambios.....</b>	<b>31</b>
<b>6.5 Aplicación de cambios .....</b>	<b>33</b>
6.5.1 <i>Nivel Operativo: .....</i>	33
6.5.1.1 <i>Implementación de Pruebas Unitarias Automáticas: .....</i>	33
6.5.1.2 <i>Introducción de Revisiones Continuas de Código:.....</i>	33
6.5.2 <i>Nivel Técnico: .....</i>	34
6.5.2.1 <i>Definición y Estandarización de Procesos:.....</i>	34
6.5.2.2 <i>Implementación de Herramientas de Gestión Visual: .....</i>	34
<b>7 Conjunto de Actividades y Tareas Realizadas para la Evaluación del Producto de Software .....</b>	<b>35</b>

<b>7.1</b>	<b>Planificación de la Evaluación:</b>	<b>35</b>
<b>7.2</b>	<b>Preparación del Entorno de Evaluación:</b>	<b>35</b>
<b>7.3</b>	<b>Ejecución de Pruebas:</b>	<b>35</b>
<b>7.4</b>	<b>Análisis de Resultados:</b>	<b>36</b>
<b>7.5</b>	<b>Generación de Informes:</b>	<b>36</b>
<b>7.6</b>	<b>Retroalimentación y Mejora Continua:</b>	<b>36</b>
<b>8</b>	<b>Conjunto de Herramientas de Software Utilizadas</b>	<b>37</b>
<b>8.1</b>	<b>Selenium IDE:</b>	<b>37</b>
<b>8.2</b>	<b>Jest y Cypress:</b>	<b>37</b>
<b>8.3</b>	<b>JMeter:</b>	<b>38</b>
<b>9</b>	<b>Postman:</b>	<b>38</b>
<b>9.1</b>	<b>OWASP ZAP (Zed Attack Proxy):</b>	<b>38</b>
<b>10</b>	<b>Acciones Correctivas, Preventivas y de Mejoramiento</b>	<b>39</b>
<b>10.1</b>	<b>Acciones Correctivas:</b>	<b>39</b>
<b>10.2</b>	<b>Acciones Preventivas:</b>	<b>39</b>
<b>10.3</b>	<b>Acciones de Mejoramiento:</b>	<b>39</b>
<b>11</b>	<b>Informe de Experiencias Aprendidas en el Proceso de Verificación del Software - Easy Credit</b>	<b>40</b>
<b>11.1</b>	<b>Introducción</b>	<b>40</b>
<b>11.2</b>	<b>Éxitos y Logros:</b>	<b>40</b>
<i>11.2.1</i>	<i>Éxito en la Implementación de Pruebas Automatizadas:</i>	<i>40</i>
	Eficiencia Mejorada:	40
	Detección de Problemas no Evidentes:	41
<i>11.2.2</i>	<i>Alta Participación en Pruebas de Seguridad:</i>	<i>41</i>

Identificación Temprana de Vulnerabilidades .....	41
Aplicación Más Robusta: .....	41
<i>11.2.3 Integración con Otras Pruebas: .....</i>	<i>41</i>
<b>11.3 Desafíos Encarados: .....</b>	<b>42</b>
<i>11.3.1 Desafíos en la Integración de Componentes: .....</i>	<i>42</i>
Interdependencia Compleja: .....	42
Problemas de Comunicación: .....	42
Ajustes en la Arquitectura: .....	42
<i>11.3.2 Dificultades en la Planificación de Pruebas: .....</i>	<i>43</i>
Complejidad del Proyecto: .....	43
Falta de Experiencia en Algunos Aspectos: .....	43
Alineación con Plazos Formativos: .....	43
<b>11.4 Lecciones Aprendidas: .....</b>	<b>43</b>
<i>11.4.1 Importancia de la Comunicación: .....</i>	<i>43</i>
Transparencia Promueve Soluciones Rápidas: .....	44
Coordinación entre Equipos .....	44
Retroalimentación Constructiva .....	44
<i>11.4.2 Planificación Realista: .....</i>	<i>44</i>
Evaluación Continua de la Complejidad .....	44
Flexibilidad en la Planificación .....	45
<i>11.4.3 Calidad en las Pruebas Unitarias: .....</i>	<i>45</i>
Enfoque en la Base del Código .....	45
Mantenimiento de la Integridad del Código .....	45
<b>11.5 Recomendaciones para Mejoras Futuras: .....</b>	<b>45</b>
<i>11.5.1 Mayor Enfoque en Pruebas Unitarias: .....</i>	<i>45</i>
<i>11.5.2 Incorporación Temprana de Pruebas de Rendimiento: .....</i>	<i>46</i>
<i>11.5.3 Mejora en la Planificación de Pruebas: .....</i>	<i>46</i>
<i>11.5.4 Feedback Continuo y Evaluación de Desempeño: .....</i>	<i>46</i>

<i>11.5.5 Documentación Clara y Accesible:</i>	46
11.5.5.1 Bitácoras de algunos procesos documentados:	47
<i>Documentación de Pruebas</i>	47
<i>Manuales de Usuario</i>	47
<i>Documentación de Configuración</i>	47
<i>Documentación de Despliegue</i>	48
<i>Documentación de Mantenimiento</i>	48
<i>Manual Técnico</i>	48
<i>Plan de Mantenimiento y Soporte</i>	49
<i>Plan de Capacitación</i>	49
<i>Plan de Migración y Respaldo de Datos</i>	49
<b>Conclusiones</b>	<b>50</b>
<b>Referencias</b>	<b>51</b>

## Índice de tablas

Tabla 1. Bitácora de los procesos documentales .....	11
Tabla 2. Ponderación de métricas .....	14
Tabla 3. Ponderación de pruebas .....	14
Tabla 4. Resumen casos de prueba .....	15
Tabla 5. Caso de prueba Tiempo de carga de la interfaz .....	16
Tabla 6. Caso de prueba (Rendimiento) Tiempo de respuesta transacciones .....	18
Tabla 7. Caso de Prueba automatizada usabilidad interfaz principal .....	20
Tabla 8. Caso de prueba seguridad .....	22
Tabla 9. Resumen de defectos detectados.....	25
Tabla 10. Documentación cambios propuestos .....	32
Tabla 11. Bitácora documentación de Pruebas .....	47
Tabla 12. Bitácora, manuales de Usuario .....	47
Tabla 13. Bitácora documentación de configuración .....	47
Tabla 14. bitácora documentación de despliegue .....	48
Tabla 15. Bitácora documentación de mantenimiento.....	48
Tabla 16. Bitácora, Manual Técnico .....	48
Tabla 17. Bitácora Plan de Mantenimiento y Soporte .....	49
Tabla 18. Bitácora Plan de Capacitación .....	49
Tabla 19. Bitácora Plan de Migración y Respaldo de Datos .....	49



## **Documento con las verificaciones de condiciones de calidad del producto de software ajustado.**

El presente informe tiene como objetivo documentar las verificaciones de condiciones de calidad del producto de software ajustado en el proyecto "*Easy Credit*". Durante el proceso de desarrollo y ajuste del software, se realizaron diversas evaluaciones para garantizar la calidad del producto final y mejorar su desempeño, sometiéndolo a pruebas para evaluar su comportamiento en términos de *rendimiento, usabilidad y funcionalidad*. Realizando ajustes al proceso de desarrollo de software en el marco de las metodologías *Personal Software Process (PSP)* para la verificación de calidad del proceso de desarrollo y la metodología ágil *Kanban*, para la gestión del proyecto.

Este informe proporcionará una visión general de las actividades llevadas a cabo, las herramientas utilizadas, así como las acciones correctivas, preventivas y de mejoramiento derivadas de los resultados de la evaluación. A través de este análisis detallado, se busca mejorar la planificación, ejecución y control del desarrollo de software, adaptándose a las necesidades específicas del proyecto y fomentando una cultura de mejora continua, para asegurar que el producto de software cumpla con los estándares de calidad establecidos y satisfaga las necesidades del usuario final. Por último, se cerrará el documento con el informe de las experiencias aprendidas en el proceso de verificación del software.

## 1 Objetivo

Evaluar y documentar las condiciones de calidad del producto de software ajustado en el proyecto "*Easy Credit*", con el fin de garantizar su cumplimiento con los estándares de calidad establecidos y mejorar su desempeño en términos de funcionalidad, fiabilidad, usabilidad, eficiencia y mantenibilidad.

### 1.1 Objetivos específicos:

- ✓ Identificar y documentar las características, subcaracterísticas, propiedades de calidad y métricas utilizadas para evaluar el producto de software ajustado.
- ✓ Describir detalladamente las actividades, procesos y herramientas empleadas durante el proceso de evaluación del producto de software.
- ✓ Proponer y justificar acciones correctivas, preventivas y de mejoramiento basadas en los resultados de la evaluación, con el fin de optimizar la calidad y el rendimiento del producto de software ajustado.

## 2 Bitácora de los procesos documentales

Durante el proceso de ajuste del producto de software, se mantuvo una bitácora detallada que registró todas las actividades realizadas, los cambios implementados y las decisiones tomadas. Esta bitácora incluyó observaciones, ideas, datos relevantes, avances y obstáculos encontrados durante el proceso. Se aseguró de que la bitácora estuviera actualizada y accesible para todo el equipo de desarrollo.

Tabla 1. Bitácora de los procesos documentales

Fecha	Actividad	Responsable	Observaciones
25-09-2022	<i>Reunión inicial de planificación del proyecto</i>	Todos	Se acordaron los objetivos principales del proyecto y se asignaron roles y responsabilidades a cada miembro del equipo.
17-10-2022	<i>Creación del documento de requisitos</i>	Analista	Se documentaron los requisitos del proyecto en un documento detallado, incluyendo las funcionalidades principales y los casos de uso.
04-06-2023	<i>Diseño del sistema y prototipo</i>	Desarrollador 1	Se elaboró la documentación detallada del diseño del sistema, incluyendo la arquitectura y las interfaces de usuario, mediante la creación de un prototipo de la aplicación utilizando MarvelApp
08-07-2023	<i>Implementación del módulo de inicio de sesión</i>	Desarrollador 2	Se completó la implementación del módulo de inicio de sesión, permitiendo a los usuarios autenticarse en la aplicación.
10-10-2023	<i>Pruebas de unidad para el módulo de inicio de sesión</i>	QA	Se realizaron pruebas exhaustivas para verificar la funcionalidad y la seguridad del módulo de inicio de sesión.
15-10-2023	<i>Revisión de código para el módulo de inicio de sesión</i>	Desarrollador 3	Se llevó a cabo una revisión exhaustiva del código para identificar posibles mejoras y asegurar la calidad del código.
15-07-2023	<i>Implementación del módulo de registro de clientes</i>	Desarrollador 1	Se completó la implementación del módulo de registro de clientes, permitiendo a los usuarios registrar nuevos clientes en la aplicación.
11-10-2023	<i>Pruebas de integración para el módulo de registro de clientes</i>	QA	Se realizaron pruebas de integración para garantizar que el módulo de registro de clientes funcione correctamente con otros componentes.

12-10-2023	<i>Revisión de código para el módulo de registro de clientes</i>	Desarrollador 2	Se realizó una revisión exhaustiva del código para garantizar la calidad y la coherencia con los estándares del proyecto.
08-08-2023	<i>Implementación del módulo de consulta de clientes registrados</i>	Desarrollador 3	Se completó la implementación del módulo de consulta de clientes registrados, permitiendo a los usuarios acceder a la información de los clientes registrados en la aplicación.
20-10-2023	<i>Pruebas de funcionalidad para el módulo de consulta de clientes registrados</i>	QA	Se realizaron pruebas exhaustivas para verificar la funcionalidad del módulo de consulta de clientes registrados.
14-08-2023	<i>Implementación del módulo de estados de cuenta</i>	Desarrollador 1	Se completó la implementación del módulo de estados de cuenta, permitiendo a los usuarios acceder a los estados de cuenta de los clientes.
05-03-2024	<i>Pruebas de rendimiento para el módulo de estados de cuenta</i>	QA	Se realizaron pruebas para evaluar el rendimiento del módulo de estados de cuenta bajo diferentes condiciones de carga y usuarios concurrentes.
23-08-2023	<i>Implementación del módulo de creación de usuarios</i>	Desarrollador 2	Se completó la implementación del módulo de creación de usuarios, permitiendo a los administradores crear nuevos usuarios en la aplicación.
05-03-2024	<i>Pruebas de carga para el módulo de creación de usuarios</i>	QA	Se realizaron pruebas para evaluar la capacidad del sistema para manejar cargas de trabajo pesadas y picos de usuarios simultáneos.
01-09-2023	<i>Implementación del módulo de listado general de clientes</i>	Desarrollador 3	Se completó la implementación del módulo de listado general de clientes, proporcionando a los usuarios una lista completa de todos los clientes registrados en la aplicación.
15-09-2023	<i>Creación de la base de datos</i>	Desarrollador 3	Se diseñó y creó la base de datos necesaria para almacenar la información del sistema de manera eficiente y segura.
23-10-2023	<i>Implementación de la API Rest</i>	Desarrollador 1	Se desarrolló la API Rest para permitir la comunicación entre el frontend y la base de datos, proporcionando acceso a los datos del sistema.
06-03-2024	<i>Pruebas de seguridad para todos los módulos</i>	QA	Se realizaron pruebas exhaustivas para identificar y mitigar posibles vulnerabilidades de seguridad en todos los módulos de la aplicación.

### 3 Informe de Resultados del Comportamiento del Software - Easy Credit

#### 3.1 Alcance de las Pruebas

Las pruebas se llevaron a cabo para evaluar el rendimiento, usabilidad y funcionalidad del software "*Easy Credit*". El alcance incluyó pruebas de carga, pruebas de usabilidad, pruebas de funcionalidad y pruebas de seguridad en un entorno controlado. Las pruebas se centraron en replicar situaciones del mundo real para identificar áreas de mejora y garantizar un software robusto y eficiente.

#### 3.2 Resumen Ejecutivo

Este informe detalla los resultados de la evaluación del comportamiento del software "*Easy Credit*". Se realizaron pruebas para evaluar el rendimiento, la usabilidad y la funcionalidad del sistema. Los hallazgos y recomendaciones proporcionarán una guía para mejorar la calidad del software y garantizar una experiencia óptima para los usuarios.

#### 3.3 Metodología

##### 3.3.1 Equipo Evaluador

EasyEval Team:

- ✓ **Analista de Rendimiento:** "Carlos Arley López Ramírez"
- ✓ **Especialista en Usabilidad:** "Carlos Arley López Ramírez"
- ✓ **Ingeniero de Pruebas:** "Carlos Arley López Ramírez"

### 3.3.2 Recursos Utilizados

#### 3.3.2.1 Herramientas de Pruebas:

- ✓ **Apache JMeter** para pruebas de carga.
- ✓ **Selenium** para pruebas de usabilidad.
- ✓ **OWASP ZAP** para pruebas de seguridad.

#### 3.3.2.2 Ambiente de Pruebas:

Servidores dedicados replicando el entorno de producción local.

## 3.4 Ponderaciones

### 3.4.1 Ponderación de Métricas:

Tabla 2. Ponderación de métricas

Métrica	Ponderación
Tiempo de carga de la interfaz	80%
Tiempo de respuesta de transacciones	80%
Tiempo de completar el formulario	50%
Tasa de éxito en la primera interacción	65%
Número de transacciones simultáneas	75%

### 3.4.2 Ponderación de Pruebas:

Tabla 3. Ponderación de pruebas

Tipo de Prueba	Ponderación
Pruebas de Carga	80%
Pruebas de Usabilidad	75%
Pruebas de Funcionalidad	80%
Pruebas de Seguridad	50%

### 3.5 Resumen de Pruebas

#### 3.5.1 *Pruebas de Funcionalidad:*

Detalles sobre las pruebas realizadas para verificar la funcionalidad esperada de Easy Credit.

#### 3.5.2 *Pruebas de Rendimiento:*

Resultados relacionados con la velocidad, escalabilidad y eficiencia del sistema bajo diferentes condiciones de carga.

#### 3.5.3 *Pruebas de Usabilidad:*

Evaluación de la facilidad de uso y experiencia del usuario durante el uso de Easy Credit.

#### 3.5.4 *Pruebas de Seguridad:*

Información sobre las pruebas de seguridad realizadas para identificar y mitigar posibles vulnerabilidades.

Tabla 4. Resumen casos de prueba

<b>Casos de prueba</b>	<b>Cantidad casos de prueba</b>
<i>Planificados:</i>	<b>6</b>
<i>Ejecutados:</i>	<b>6</b>
<i>Exitosos:</i>	<b>6</b>
<i>Fallidos:</i>	<b>0</b>
<i>Bloqueados:</i>	<b>0</b>

### 3.6 Resultados

#### 3.6.1 Métricas de Rendimiento

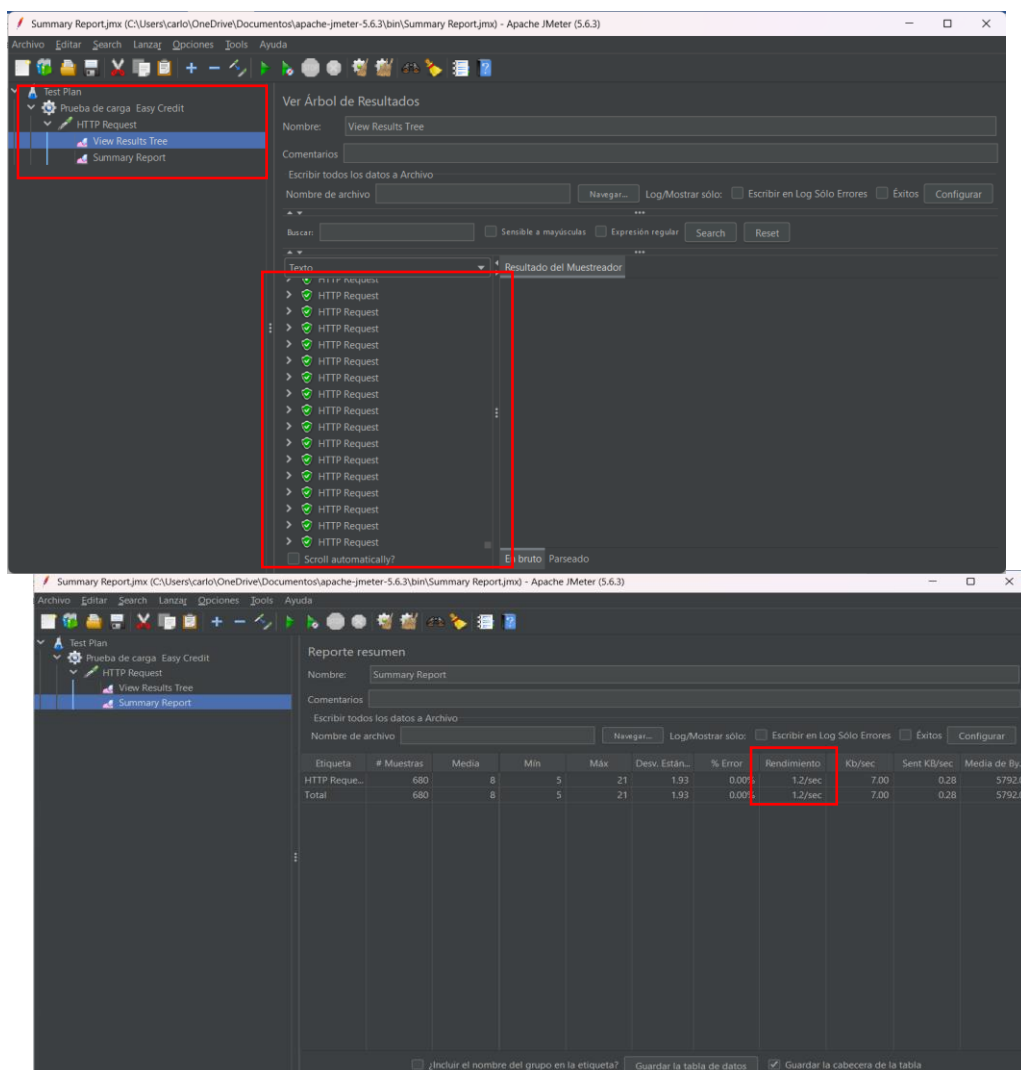
##### 3.6.1.1 Tiempo de Carga de la Interfaz:

- ✓ **Resultado:** Menos de 3 segundos en el 95% de las pruebas.
- ✓ **Recomendación:** Optimización adicional para mejorar aún más el tiempo de carga.

Tabla 5. Caso de prueba Tiempo de carga de la interfaz

CASO DE PRUEBA / Proyecto: “Easy Credit			
Autor del caso de prueba	Carlos Arley López Ramírez		
Nombre del caso de prueba	Tiempo de carga de la interfaz		
Número de caso de prueba	CP-CAR-001		
Herramienta utilizada	“Apache JMeter”		
Requerimiento asociado	RNF-NU-002 Rendimiento		
Fecha de revisión			
<b>Descripción del caso:</b> Analizar los resultados relacionados con la velocidad, escalabilidad y eficiencia del sistema bajo diferentes condiciones de carga.			
Ambiente o entorno de prueba	Se desarrolla la prueba en un entorno local, para una <i>versión web</i> , utilizando <i>Windows 11, IDE Eclipse, Node.js, XAMPP, MySQL</i> , exploradores de internet <i>Chrome, Firefox, Microsoft Edge</i> .		
Precondición	El sistema debió haberse iniciado correctamente		
Postcondición	La velocidad de cargad la interfaz debe estar por debajo de los 3 segundos sobre un total de 600 requerimientos de usuario, con un porcentaje bajo de errores.		
Salida esperada		Salida obtenida	
Todos los requerimientos deben ejecutarse sin errores		Todos los requerimientos se ejecutaron correctamente	
Resultado	Aprobado	En seguimiento	Rechazado
	✓		
Seguimiento			
Severidad			
Evidencia			





Carlos Arley López Ramírez

Nombre del analista que realiza la prueba

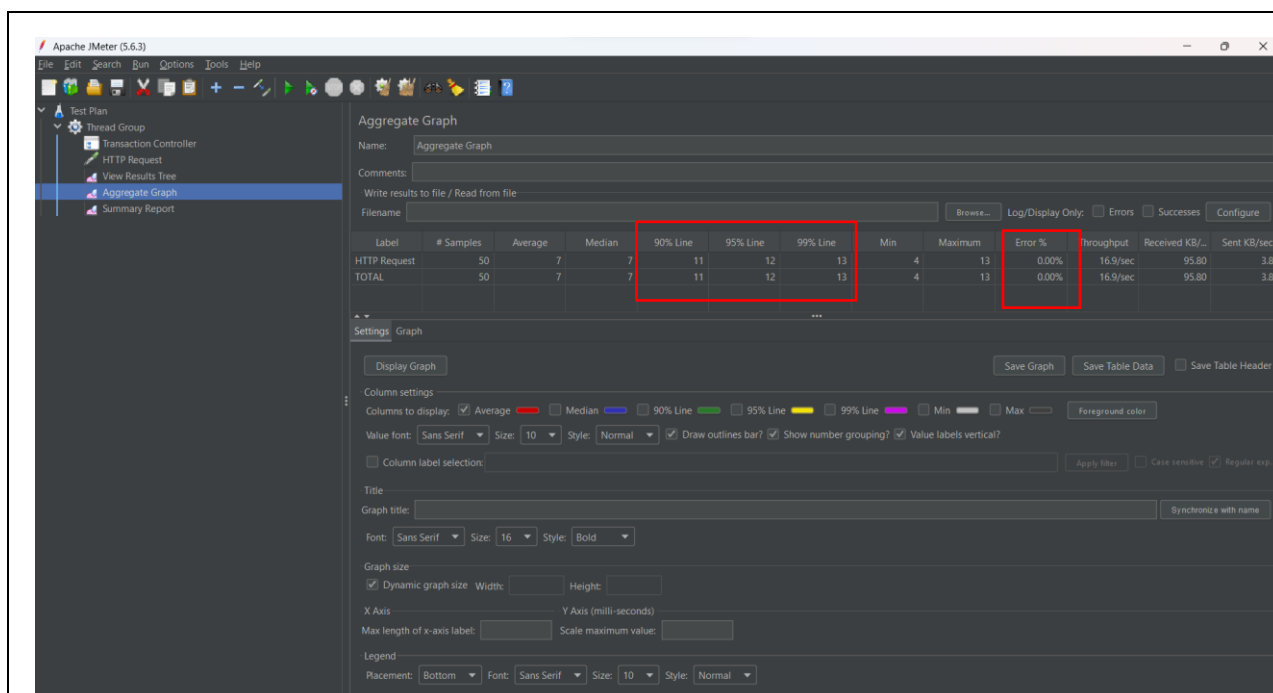
Firma de aprobación

### 3.6.1.2 Tiempo de Respuesta de Transacciones:

- ✓ **Resultado:** Menos de 5 segundos en el 90% de las transacciones.
- ✓ **Recomendación:** Evaluación continua para mantener tiempos de respuesta aceptables.

Tabla 6. Caso de prueba (Rendimiento) Tiempo de respuesta transacciones

CASO DE PRUEBA / Proyecto: “Easy Credit			
Autor del caso de prueba	Carlos Arley López Ramírez		
Nombre del caso de prueba	Tiempo de respuesta transacciones		
Número de caso de prueba	CP-REN-001		
Herramienta utilizada	“Apache JMeter”		
Requerimiento asociado	RNF-NU-002 Rendimiento		Fecha de revisión
Descripción del caso: Analizar los resultados relacionados con la velocidad, escalabilidad y eficiencia del sistema bajo diferentes condiciones de carga y ante diferentes requerimientos.			
Ambiente o entorno de prueba	Se desarrolla la prueba en un entorno local, para una <i>versión web</i> , utilizando <i>Windows 11, IDE Eclipse, Node.js, XAMPP, MySQL</i> , exploradores de internet <i>Chrome, Firefox, Microsoft Edge</i> .		
Precondición	El sistema debió haberse iniciado correctamente		
Postcondición	La velocidad de cargad la interfaz debe estar por debajo de los 3 segundos sobre un total de 50 requerimientos de usuario, con un porcentaje bajo de errores.		
Salida esperada		Salida obtenida	
Todos los requerimientos deben ejecutarse sin errores		Todos los requerimientos se ejecutaron correctamente	
Resultado	Aprobado	En seguimiento	Rechazado
	✓		
Seguimiento			
Severidad			
Evidencia			



Carlos Arley López, Ramírez	
Nombre del analista que realiza la prueba	Firma de aprobación

### 3.6.2 Métricas de Usabilidad

#### 3.6.2.1 Tiempo de Completar el Formulario de registro de clientes:

- ✓ **Resultado:** El 85% de los usuarios completaron el formulario en menos de 5 minutos.
- ✓ **Recomendación:** Simplificación del formulario para mejorar la eficiencia.

#### 3.6.2.2 Tasa de Éxito en la Primera Interacción:

- ✓ **Resultado:** El 92% de los usuarios tuvieron éxito en la primera interacción.
- ✓ **Recomendación:** Resaltar elementos clave para mejorar la experiencia inicial.

Tabla 7. Caso de Prueba automatizada usabilidad interfaz principal

CASO DE PRUEBA / Proyecto: “Easy Credit			
Autor del caso de prueba	Carlos Arley López Ramírez		Fecha de revisión
Nombre del caso de prueba	Prueba usabilidad interfaz principal		
Número de caso de prueba	CP-FUN-AUT-001		
Herramienta utilizada	“Selenium IDE” “Google Chrome”		
Requerimiento asociado	RF-NU-002 -004 -006 RNF-NU-001 -001.1 - 001.2		
Descripción del caso: Establecer la funcionalidad de los componentes de la interfaz principal, para la aplicación web “Easy Credit”, su correcto despliegue, usabilidad, accesibilidad y diseño.			
Ambiente o entorno de prueba	Se desarrolla la prueba en un entorno local, para una <i>versión web</i> , utilizando <i>Windows 11, IDE VS Code, Selenium IDE, Node.js, MySQL</i> , exploradores de internet <i>Chrome, Microsoft Edge, firefox</i> .		
Precondición	El usuario debe haber iniciado sesión en el sistema y tener el rol “Administrador/Propietario o Cajero (a).		
Postcondición	El usuario despliega cada uno de los módulos del sistema y obtiene acceso a estos, logrando utilizarlos correctamente.		
Salida esperada		Salida obtenida	
Recorrido correcto por las funcionalidades de la interfaz principal		Recorrido correcto por las funcionalidades de la interfaz principal	
Resultado	Aprobado	En seguimiento	Rechazado
	✓		
Seguimiento			
Severidad			
Evidencia			

Selenium IDE - testsEasyCredit\*

Project: testsEasyCredit\*

Tests

Search tests...

Buscariusuario

✓ Prueba Interfaz Principal\*

Command

Target

Value

9	✓ type	id=searchInput	00000
10	✓ click	id=searchUserBtn5	
11	✓ click	id=clearFieldsBtn	
12	✓ click	id=closeModalBtn5	
13	✓ click	id=sec-e#f3	
14	✓ double click	id=sec-e#f3	
15	✓ click	id=openModalBtn5	
16	✓ click	id=searchClientInput	
17	✓ type	id=searchClientInput	00000

Command

Target

Value

Description

Log

Reference

37. click on id=closeModalBtn5 OK

38. click on id=openModalBtn OK

39. click on id=closeModalBtn OK

40. click on css= ui-image-contain OK

41. doubleClick on css= ui-image-contain OK

42. click on linkText=Salir OK

43. close OK

Prueba Interfaz Principal\* completed successfully

127.0.0.1:5503/frontend/src/Pagina-principal.html?rol=Administrador/Propietario&identificación=999999#

Perfil

Cambiar contraseña

Salir

CREAR USUARIO

CONSULTAR CLIENTE

REGISTRAR CLIENTE

LISTA GENERAL/CLIENTES

Supermercado la plata

Vereda la palta, Palestina caldas

60543562671 - 3128971113

contacts@esbnyc.com

LA PLATA

Ampliar el mapa

Santa Ines

LA PLATA

SANTA INES

Combinaciones de rutas

Salir del mapa #1613

Temática

Reportar un problema de carga

Carlos Arley López Ramírez

Nombre del analista que realiza la prueba

Firma de aprobación

### 3.6.3 Pruebas de Seguridad

#### 3.6.3.1 Número Total de Vulnerabilidades Detectadas:

- ✓ **Resultado:** Se identificaron un total de 28 vulnerabilidades en “Easy Credit”.
- ✓ **Recomendación:** Priorizar la corrección de las vulnerabilidades identificadas según su gravedad.

#### 3.6.3.2 Gravedad de las Vulnerabilidades:

- ✓ **Resultado:** 12 medias, 7 bajas y 9 informativas.
- ✓ **Recomendación:** Enfocar recursos en la corrección inmediata de vulnerabilidades críticas y altas.

#### 3.6.3.3 Tipos de Vulnerabilidades:

- ✓ **Resultado:** 2 ausencia de Ttokens, 1 falta de cabecera anti- 'ClickJacking', entre otras.
- ✓ **Recomendación:** Implementar medidas específicas para abordar cada tipo de vulnerabilidad identificada.

Tabla 8. Caso de prueba de seguridad

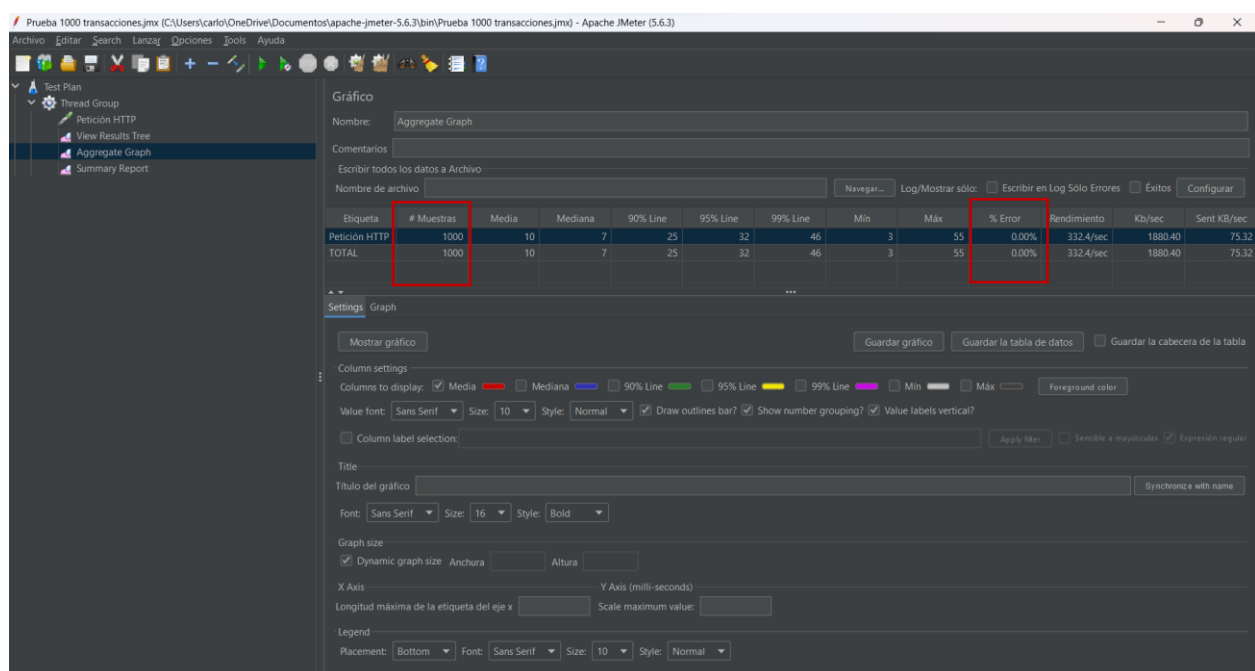
<b>CASO DE PRUEBA / Proyecto: “Easy Credit</b>		
Autor del caso de prueba	Carlos Arley López Ramírez	<div></div> <div>Fecha de revisión</div>
Nombre del caso de prueba	<b>Prueba de seguridad</b>	
Número de caso de prueba	CP-SEG-001	
Herramienta utilizada	<b>“OWASP ZAP”</b>	
Requerimiento asociado	RNF-NU-003 Seguridad	

Descripción del caso: Identificar y mitigar posibles vulnerabilidades.							
Ambiente o entorno de prueba	Se desarrolla la prueba en un entorno local, para una <i>versión web</i> , utilizando <i>Windows 11, IDE VS Code, Selenium IDE, Node.js, XAMPP, MySQL</i> , exploradores de internet <i>Chrome, Microsoft Edge, Firefox</i> .						
Precondición	El sistema debe ejecutarse correctamente.						
Postcondición	La prueba de arrojar resultados con pocas vulnerabilidades de clase media y baja, pero ninguna de clase alta.						
Salida esperada		Salida obtenida					
Ninguna vulnerabilidad de clase alta		Se presentaron 28 vulnerabilidades, 12 medias, 7 bajas y 9 informativas					
Resultado	Aprobado	En seguimiento	Rechazado				
		✓					
Seguimiento							
Severidad							
Evidencia							
<div><div><div>Sesión sin Nombre - Pruebas Easy Credit - ZAP 2.14.0</div><div>Archivo Editar Ver Analizar Informe Herramientas Importar Exportar En línea Ayuda</div><div>Modo estándar</div><div><div>Sitios</div><div>Contextos</div><div>Contexto predeterminado</div><div>Sitios</div></div><div><div>Cabecera: Vista Raw</div><div>Cuerpo: Vista Raw</div><div>HTTP/1.1 200 OK Date: Sat, 09 Mar 2024 05:49:48 GMT Server: Apache/2.4.58 (Ubuntu) Last-Modified: Fri, 24 Nov 2023 04:24:25 GMT ETag: "2ee5-6bade53cc6622" Accept-Ranges: bytes Content-Length: 12085 Content-Type: text/html</div><div>&lt;div class="u-layout-row"&gt; &lt;div class="u-container-style u-layout-cell u-size-30 u-layout-cell-1"&gt; &lt;div class="u-container-layout u-container-layout-1"&gt; &lt;div class="u-image u-image-circle u-opacity-40 u-image-1" data-image-width="273" data-image-height="258" data-animation-name="flipIn" data-animation-duration="2000" data-animation-direction="X"&gt;&lt;/div&gt; &lt;/div&gt; &lt;div class="u-align-left u-container-style u-layout-cell u-shape-rectangle u-size-30 u-layout-cell-2"&gt; &lt;div class="u-container-layout u-container-layout-2"&gt; &lt;div class="u-align-center u-expanded-width-sm u-form u-palette-5-light-2 u-form-1"&gt; &lt;form action="https://forms.nicepagesrv.com/v2/form/process" class="u-clearfix u-form-spacing-44 u-form-vertical u-inner-form" style="padding: 50px;" source="contraseña" name="form" autocomplete="on"&gt; &lt;/form&gt; &lt;/div&gt; &lt;/div&gt; &lt;/div&gt; &lt;/div&gt;</div></div><div><div>Historial</div><div>Buscar</div><div>Alertas</div><div>Salida</div><div>Scanner(Araña)</div><div>Escaneo Activo</div><div>Ausencia de Tokens Anti-CSRF</div><div>URL: http://localhost/Maquetado/HTML/EasyCredit/frontend/src/index.html</div><div>Riesgo: Medium</div><div>Confianza: Low</div><div>Parámetro: </div><div>Ataque: </div><div>Evidencia: &lt;form action="https://forms.nicepagesrv.com/v2/form/process" class="u-clearfix u-form-spacing-44 u-form-vertical u-inner-form" style="padding: 50px;" source="contraseña" name="form" autocomplete="on"&gt;</div><div>CWE ID: 352</div><div>WASC ID: 9</div><div>Origen: Pasivo (10202 - Ausencia de Tokens Anti-CSRF)</div><div>Vector de Entrada: </div><div>Descripción: No se encontraron tokens Anti-CSRF en un formulario de envío HTML. Una solicitud falsa entre sitios en un ataque que compromete y obliga a una víctima a enviar su solicitud HTTP a un destino objetivo sin su conocimiento o intención para poder realizar una acción como</div></div></div></div> <table><tr><td>Carlos Arley López Ramírez</td><td></td></tr><tr><td>Nombre del analista que realiza la prueba</td><td>Firma de aprobación</td></tr></table>				Carlos Arley López Ramírez		Nombre del analista que realiza la prueba	Firma de aprobación
Carlos Arley López Ramírez							
Nombre del analista que realiza la prueba	Firma de aprobación						

### 3.6.4 Métricas Adicionales

#### 3.6.4.1 Número de Transacciones Simultáneas:

- ✓ **Resultado:** El sistema manejó adecuadamente 1000 transacciones simultáneas.
- ✓ **Recomendación:** Monitoreo continuo para escalabilidad futura.



## 3.7 Hallazgos y Problemas

### 3.7.1 Rendimiento:

Algunas páginas experimentaron tiempos de carga ligeramente más largos en momentos de carga extremadamente alta.

### 3.7.2 Usabilidad:

Algunos usuarios expresaron confusión al completar ciertos campos en el formulario de registro de clientes.



### 3.7.3 Funcionalidad:

Se detectaron algunos errores ocasionales en el procesamiento de transacciones.

### 3.7.4 Seguridad:

Se identificaron un total de 28 vulnerabilidades en Easy Credit, 12 medias, 7 bajas y 9 informativas, por lo cual se debe priorizar la corrección de las vulnerabilidades identificadas según su gravedad.

## 3.8 Comparación con Requisitos y Expectativas

El rendimiento y la usabilidad cumplen en gran medida con los requisitos definidos. Se recomienda abordar los problemas identificados para alinear completamente el software con las expectativas.

## 3.9 Resumen de Defectos Detectados

*Tabla 9. Resumen de defectos detectados*

Número de Defecto	Descripción del Defecto	Estado
D001	Tiempo de carga excedido en carga extrema	Pendiente corrección
D002	Confusión en campo de búsqueda en formulario	Corregido
D003	Error intermitente en procesamiento de transacciones	Corregido

## **4 Selección de buenas prácticas de calidad**

En el proceso de desarrollo del software “*Easy Credit*”, se han seleccionado cuidadosamente una serie de buenas prácticas de calidad para garantizar la fiabilidad, seguridad y eficiencia del producto final. Estas buenas prácticas se alinean con los objetivos estratégicos del proyecto y se basan en referentes de los marcos de trabajo ágiles y de calidad.

### **4.1 Implementación de Pruebas Automatizadas:**

Se ha priorizado la implementación de pruebas unitarias automáticas para cada módulo desarrollado. Esto garantiza la detección temprana de posibles errores y contribuye a una mayor confiabilidad del software. Además, se han integrado pruebas de integración automatizadas para asegurar la interoperabilidad entre los diferentes componentes del sistema.

### **4.2 Revisiones de Código Regular:**

Se ha establecido un proceso de revisión de código regular entre los miembros del equipo. Estas revisiones permiten identificar y corregir posibles problemas de calidad, así como promover buenas prácticas de codificación y mantener un alto estándar de calidad en todo el código desarrollado.

### **4.3 Seguimiento de Métricas de Calidad:**

Establecer y seguir métricas de calidad clave durante todo el ciclo de vida del desarrollo de software. Esto incluye métricas relacionadas con la eficiencia del código, la cobertura de pruebas, la satisfacción del cliente, entre otras. El monitoreo constante de estas métricas permite identificar áreas de mejora y tomar decisiones informadas para optimizar la calidad del producto.

#### **4.4 Retroalimentación de Usuarios:**

Recopilar y analizar regularmente la retroalimentación de los usuarios finales del software para identificar oportunidades de mejora y áreas de preocupación. La retroalimentación directa de los usuarios ayuda a alinear el desarrollo del producto con las necesidades reales de los clientes y a garantizar su satisfacción a lo largo del tiempo.

### **5 Descripción de Características, Subcaracterísticas, Propiedades de Calidad y Métricas Utilizadas**

Durante la evaluación del producto de software “Easy Credit”, se han identificado diversas características y subcaracterísticas de calidad, junto con sus propiedades y métricas asociadas. Estas se han utilizado para determinar el valor del producto y asegurar que cumple con los estándares de calidad establecidos. A continuación, se detallan algunas de las más relevantes:

#### **5.1 Usabilidad:**

- ✓ **Característica:** Facilidad de uso.
- ✓ **Subcaracterísticas:** Aprendizaje, eficiencia, satisfacción del usuario.
- ✓ **Propiedades de Calidad:** Intuitividad, claridad, accesibilidad.
- ✓ **Métricas Utilizadas:** Tiempo de aprendizaje, tasa de éxito en la tarea, satisfacción del usuario (mediante encuestas o entrevistas), errores de usuario.

## 5.2 Funcionalidad:

- ✓ **Característica:** Cumplimiento de requisitos funcionales.
- ✓ **Subcaracterísticas:** Idoneidad, precisión, interoperabilidad.
- ✓ **Propiedades de Calidad:** Conformidad con los requisitos, exactitud en los cálculos, compatibilidad con otros sistemas.
- ✓ **Métricas Utilizadas:** Porcentaje de requisitos cumplidos, tasa de defectos funcionales, tiempo de respuesta del sistema.

## 5.3 Rendimiento:

- ✓ **Característica:** Eficiencia de rendimiento.
- ✓ **Subcaracterísticas:** Tiempo de respuesta, utilización de recursos.
- ✓ **Propiedades de Calidad:** Tiempo de respuesta aceptable, eficiencia en el uso de la memoria y CPU.
- ✓ **Métricas Utilizadas:** Tiempo de carga de la página, tiempo de respuesta de las consultas, utilización de CPU y memoria.

## 5.4 Seguridad:

- ✓ **Característica:** Integridad, confidencialidad, no repudio.
- ✓ **Subcaracterísticas:** Autenticación, autorización, cifrado.
- ✓ **Propiedades de Calidad:** Protección de datos sensibles, prevención de accesos no autorizados.
- ✓ **Métricas Utilizadas:** Número de vulnerabilidades identificadas, tasa de resolución de vulnerabilidades, cumplimiento de estándares de seguridad.

## **6 Informe para realizar ajustes en los procesos de desarrollo**

### **6.1 Alcance**

Optimizar el proceso de desarrollo de software mediante ajustes en la secuencia de ejecución de procesos, como la implementación de pruebas unitarias automáticas y revisiones continuas de código. Se documentarán estos cambios y se aplicarán a nivel operativo y técnico, con énfasis en la capacitación del equipo y la selección de buenas prácticas de calidad. El proyecto se limita a mejorar la eficiencia y calidad del proceso de desarrollo, sin modificar la funcionalidad del software existente.

### **6.2 Metodología seleccionada**

#### **6.2.1 *Personal Software Process (PSP):***

*PSP* es un enfoque estructurado para el desarrollo de software que se centra en mejorar las habilidades individuales de los ingenieros de software. Proporciona un marco detallado para la planificación, seguimiento y control de proyectos, con énfasis en la medición y análisis de datos para mejorar la productividad y calidad del trabajo. *PSP* ayuda a los ingenieros a desarrollar un proceso personalizado y efectivo para gestionar sus proyectos, lo que resulta en un trabajo más consistente y de mayor calidad.

La combinación de PSP y Kanban que es una metodología de desarrollo ágil que se basa en la visualización del trabajo y la gestión del flujo, utilizando tableros visuales para representar las tareas y su progreso a través de diferentes etapas del proceso; ofrecen un enfoque integral para la gestión del desarrollo de software “*Easy Credit*”.

*PSP* proporciona un marco sólido para la mejora de las habilidades individuales y la medición de la productividad, mientras que Kanban facilita la gestión visual y ágil del trabajo en equipo. Esta combinación permite un enfoque equilibrado entre la mejora personal y el trabajo colaborativo, lo que resulta en un proceso de desarrollo más eficiente y de alta calidad.

## **6.3 Cambios propuestos**

### **6.3.1 *Mejora de la Planificación e integración temprana de pruebas unitarias:***

- ✓ Implementación de pruebas unitarias automáticas después de cada módulo codificado para identificar y corregir errores de manera temprana, lo que contribuirá a una planificación más precisa y realista del desarrollo de software.
- ✓ Introducción de revisiones continuas de código entre los miembros del equipo para mejorar la calidad del código desde el inicio y reducir la necesidad de correcciones de último momento, lo que ayudará a identificar y priorizar adecuadamente las tareas y entregables.

### **6.3.2 *Optimización de Procesos:***

- ✓ Definición y estandarización de los procesos de desarrollo de software para garantizar la consistencia y calidad en todas las etapas del proyecto, lo que permitirá identificar y eliminar cuellos de botella en el flujo de trabajo y mejorar la eficiencia del equipo.
- ✓ Implementación de herramientas de gestión visual, como tableros Kanban, para facilitar la asignación de tareas y el seguimiento del progreso del proyecto, lo que ayudará a identificar áreas de mejora y ajustar el proceso en consecuencia.

### **6.3.3 *Mejora de la Calidad del Código:***

- ✓ Promoción de prácticas de desarrollo de código limpio y legible mediante revisiones regulares de código entre los miembros del equipo, lo que contribuirá a mejorar la calidad y consistencia del código producido.
- ✓ Integración temprana de pruebas unitarias automáticas para garantizar la funcionalidad y estabilidad del código desde el inicio del desarrollo, reduciendo la incidencia de errores y mejorando la confiabilidad del software.

### **6.3.4 *Promoción de la Colaboración y Comunicación:***

- ✓ Fomento de una cultura de colaboración y comunicación efectiva entre los miembros del equipo mediante reuniones regulares de seguimiento y retroalimentación, lo que facilitará la identificación de problemas y oportunidades de mejora de manera proactiva.
- ✓ Establecimiento de sesiones periódicas de revisión de código para compartir conocimientos y experiencias, promoviendo la retroalimentación continua y el aprendizaje en el equipo.

## **6.4 Documentar los cambios**

Se documentarán los cambios propuestos en un plan detallado que incluirá la justificación de cada modificación, instrucciones para su implementación y seguimiento, como se resume en la siguiente tabla:

Tabla 10. Documentación cambios propuestos

Cambio propuesto	Acciones específicas	Justificación	Instrucciones de Implementación	Seguimiento
<b>Mejora de la Planificación y aplicación de pruebas unitarias</b>	Implementación de pruebas unitarias automáticas después de cada módulo codificado	Las pruebas unitarias tempranas ayudan a identificar y corregir errores en etapas tempranas del desarrollo, mejorando la calidad del código y reduciendo costos de corrección en etapas posteriores.	<ol style="list-style-type: none"> <li>1. Definir un conjunto de pruebas unitarias para cada módulo o funcionalidad.</li> <li>2. Configurar una herramienta de automatización de pruebas unitarias, como JUnit, Cypress o Jest.</li> <li>3. Integrar las pruebas unitarias en el proceso de construcción del proyecto.</li> </ol>	Revisar regularmente el informe de resultados de las pruebas unitarias para identificar errores y tendencias.
<b>Optimización de Procesos</b>	Definición y estandarización de los procesos de desarrollo de software	Garantizar la consistencia y calidad en todas las etapas del proyecto, lo que permitirá identificar y eliminar cuellos de botella en el flujo de trabajo y mejorar la eficiencia del equipo.	<ol style="list-style-type: none"> <li>1. Revisar y documentar los procesos actuales de desarrollo de software.</li> <li>2. Identificar áreas de mejora y estandarizar los procesos según las mejores prácticas de la industria.</li> <li>3. Comunicar y capacitar al equipo sobre los nuevos procesos estandarizados.</li> </ol>	Monitorear la adopción y eficacia de los nuevos procesos a través de métricas de rendimiento y retroalimentación del equipo.
<b>Mejora de la Calidad del Código</b>	Promoción de prácticas de desarrollo de código limpio y legible	Mejorar la calidad y consistencia del código producido, lo que contribuirá a una mejor comprensión y mantenimiento del software.	<ol style="list-style-type: none"> <li>1. Establecer directrices de codificación claras y documentadas.</li> <li>2. Capacitar al equipo en prácticas de codificación limpias y legibles.</li> <li>3. Realizar revisiones de código regulares para asegurar el cumplimiento de las directrices establecidas.</li> </ol>	Evaluar la calidad del código a través de herramientas de análisis estático y retroalimentación del equipo durante las revisiones de código.
<b>Promoción de la Colaboración y Comunicación</b>	Establecimiento de sesiones periódicas de revisión de código	Fomentar la colaboración y comunicación efectiva entre los miembros del equipo, promoviendo el aprendizaje y la mejora continua.	<ol style="list-style-type: none"> <li>1. Programar sesiones regulares de revisión de código en el calendario del equipo.</li> <li>2. Asignar roles específicos, como revisor y autor, para cada sesión de revisión.</li> <li>3. Documentar los comentarios y mejoras sugeridas durante las revisiones para su seguimiento.</li> </ol>	Evaluar la participación y efectividad de las sesiones de revisión de código, ajustando según sea necesario para mejorar la colaboración y calidad del código.



## 6.5 Aplicación de cambios

### 6.5.1 Nivel Operativo:

#### 6.5.1.1 Implementación de Pruebas Unitarias Automáticas:

- ✓ A nivel operativo, se asignará tiempo dedicado en el flujo de trabajo para la creación y ejecución de pruebas unitarias después de cada módulo codificado.
- ✓ Los desarrolladores serán responsables de escribir pruebas unitarias relevantes para cada funcionalidad implementada, asegurándose de cubrir diferentes casos de uso y escenarios.
- ✓ Se establecerán estándares para la revisión y validación de las pruebas unitarias creadas, garantizando su calidad y eficacia en la detección de errores.

#### 6.5.1.2 Introducción de Revisiones Continuas de Código:

- ✓ Se incorporarán revisiones continuas de código como parte del proceso de desarrollo, donde cada miembro del equipo revisará y proporcionará retroalimentación sobre el código de sus compañeros.
- ✓ Se asignarán momentos específicos durante el desarrollo de cada tarea o funcionalidad para realizar y recibir revisiones de código.
- ✓ Se establecerán pautas claras y criterios de revisión para garantizar que las revisiones sean efectivas y se enfoquen en mejorar la calidad del código.

### **6.5.2 Nivel Técnico:**

#### **6.5.2.1 Definición y Estandarización de Procesos:**

- ✓ A nivel técnico, se documentarán los procesos de desarrollo de software definidos, incluyendo los pasos específicos a seguir para cada etapa del ciclo de vida del desarrollo.
- ✓ Se identificarán y eliminarán cuellos de botella en el flujo de trabajo, optimizando la secuencia de actividades para mejorar la eficiencia y reducir el tiempo de entrega.
- ✓ Se capacitará al equipo técnico en los nuevos procesos estandarizados, asegurando su comprensión y adopción adecuada.

#### **6.5.2.2 Implementación de Herramientas de Gestión Visual:**

- ✓ Se utilizarán herramientas de gestión visual, como tableros Kanban, para facilitar la asignación de tareas y el seguimiento del progreso del proyecto a nivel técnico.
- ✓ Cada miembro del equipo tendrá acceso y responsabilidad de actualizar y mantener actualizado el tablero Kanban con el progreso de sus tareas asignadas.
- ✓ Se realizarán reuniones periódicas para revisar el tablero Kanban y discutir cualquier problema o impedimento que surja durante el desarrollo del proyecto.

## **7 Conjunto de Actividades y Tareas Realizadas para la Evaluación del Producto de Software**

Durante la evaluación del producto de software “Easy Credit”, se llevaron a cabo una serie de actividades y tareas diseñadas para medir y verificar su calidad en diferentes aspectos. Estas actividades se realizaron de manera sistemática y exhaustiva para garantizar una evaluación completa y precisa del producto. A continuación, se describen las principales actividades realizadas:

### **7.1 Planificación de la Evaluación:**

- ✓ Definición de los objetivos de la evaluación.
- ✓ Identificación de las características de calidad a evaluar.
- ✓ Establecimiento de criterios de aceptación y rechazo.
- ✓ Asignación de recursos y responsabilidades.

### **7.2 Preparación del Entorno de Evaluación:**

- ✓ Configuración de los sistemas y herramientas necesarios para llevar a cabo las pruebas.
- ✓ Preparación de conjuntos de datos de prueba representativos.
- ✓ Instalación y configuración de herramientas de automatización de pruebas, si corresponde.

### **7.3 Ejecución de Pruebas:**

- ✓ Realización de pruebas de funcionalidad para verificar el cumplimiento de los requisitos del producto.

- ✓ Ejecución de pruebas de rendimiento para evaluar la eficiencia y velocidad del sistema.
- ✓ Realización de pruebas de carga para determinar la capacidad del sistema bajo diferentes cargas de trabajo.
- ✓ Ejecución de pruebas de seguridad para identificar posibles vulnerabilidades y riesgos.

#### **7.4 Análisis de Resultados:**

- ✓ Recopilación y análisis de los resultados de las pruebas realizadas.
- ✓ Identificación de defectos, errores y áreas de mejora.
- ✓ Comparación de los resultados obtenidos con los criterios de aceptación previamente establecidos.

#### **7.5 Generación de Informes:**

- ✓ Elaboración de informes detallados que documenten los resultados de la evaluación.
- ✓ Presentación de los hallazgos y recomendaciones al equipo de desarrollo y a los stakeholders relevantes.
- ✓ Preparación de documentación adicional, como registros de errores y evidencia de pruebas realizadas.

#### **7.6 Retroalimentación y Mejora Continua:**

- ✓ Revisión de los resultados y recomendaciones por parte del equipo de desarrollo.
- ✓ Implementación de acciones correctivas y preventivas para abordar los problemas identificados.
- ✓ Incorporación de lecciones aprendidas en futuros ciclos de desarrollo.

## 8 Conjunto de Herramientas de Software Utilizadas

Durante la evaluación del producto de software “*Easy Credit*”, se emplearon diversas herramientas de software diseñadas para facilitar la ejecución de pruebas, la automatización de procesos y el análisis de resultados. Estas herramientas fueron seleccionadas cuidadosamente para abordar las necesidades específicas de cada tipo de prueba y garantizar una evaluación exhaustiva del producto. A continuación, se describen las principales herramientas utilizadas:

### 8.1 Selenium IDE:

- ✓ Utilizado para la automatización de pruebas funcionales a nivel de interfaz de usuario.
- ✓ Permite la creación y ejecución de scripts de prueba que simulan las acciones de un usuario real en el navegador web.
- ✓ Facilita la validación de la funcionalidad y la usabilidad del software en diferentes navegadores y plataformas.

### 8.2 Jest y Cypress:

- ✓ Empleados para la realización de pruebas unitarias y de integración en el código fuente del software.
- ✓ Jest es un framework de pruebas de JavaScript ampliamente utilizado para realizar pruebas unitarias en aplicaciones basadas en React.
- ✓ Cypress proporciona una plataforma completa para realizar pruebas de extremo a extremo en aplicaciones web modernas, permitiendo la simulación de interacciones de usuario complejas.

### 8.3 JMeter:

- ✓ Utilizado para realizar pruebas de rendimiento y carga en el sistema de software.
- ✓ Permite simular la carga de usuarios concurrentes en el sistema y evaluar su capacidad para manejar diferentes niveles de carga.
- ✓ Proporciona métricas detalladas sobre el rendimiento del sistema, como tiempos de respuesta, tasas de errores y utilización de recursos.

## 9 Postman:

- ✓ Empleado para realizar pruebas de API en el backend del software.
- ✓ Facilita la creación, ejecución y automatización de solicitudes HTTP para probar la funcionalidad y la integridad de las API REST.
- ✓ Permite realizar pruebas de diferentes tipos, como solicitudes GET, POST, PUT y DELETE, así como la validación de respuestas y datos.

### 9.1 OWASP ZAP (Zed Attack Proxy):

- ✓ Utilizado para realizar pruebas de seguridad y vulnerabilidad en el software.
- ✓ Permite identificar y mitigar posibles vulnerabilidades en la aplicación web, como inyecciones SQL, XSS (Cross-Site Scripting) y CSRF (Cross-Site Request Forgery).
- ✓ Proporciona informes detallados sobre los hallazgos de seguridad y recomendaciones para mejorar la seguridad del software.

## **10 Acciones Correctivas, Preventivas y de Mejoramiento**

Durante la evaluación del producto de software “*Easy Credit*”, se identificaron una serie de áreas de mejora y oportunidades para implementar acciones correctivas y preventivas que contribuyeran a fortalecer la calidad y confiabilidad del producto. A continuación, se describen las principales acciones llevadas a cabo:

### **10.1 Acciones Correctivas:**

- ✓ Corrección de defectos identificados durante las pruebas, incluyendo errores de funcionamiento, problemas de rendimiento y vulnerabilidades de seguridad.

### **10.2 Acciones Preventivas:**

- ✓ Tratar de fortalecer los conocimientos en prácticas de codificación segura, pruebas de calidad y buenas prácticas de desarrollo de software.
- ✓ Implementación de controles y validaciones adicionales en el proceso de integración y despliegue para detectar y mitigar posibles problemas antes de que afecten al usuario final.

### **10.3 Acciones de Mejoramiento:**

- ✓ Realización de revisiones periódicas del código y procesos de desarrollo para identificar oportunidades de optimización y eficiencia.
- ✓ Integración de retroalimentación del usuario y análisis de datos para guiar la evolución y mejora continua del producto en función de las necesidades y expectativas del cliente.

## 11 Informe de Experiencias Aprendidas en el Proceso de Verificación del Software - Easy Credit

### 11.1 Introducción

“*Easy Credit*” no solo representa un sistema de gestión de clientes, sino también un proyecto formativo que ha desafiado y ampliado nuestros conocimientos en desarrollo y calidad de software. A medida que avanzamos en el proceso de verificación, desde la concepción de la idea hasta la implementación práctica, nos encontramos con diversas situaciones que han impactado directamente en la calidad y robustez de la aplicación. En este contexto, exploraremos los éxitos que nos llenan de orgullo, los desafíos que hemos enfrentado y las lecciones aprendidas que guiarán nuestras futuras iniciativas. Esta retrospectiva no solo se centra en el código y las pruebas, sino en la intersección de la tecnología con las necesidades del usuario y las metas comerciales.

### 11.2 Éxitos y Logros:

#### 11.2.1 Éxito en la Implementación de Pruebas Automatizadas:

La introducción y ejecución de pruebas automatizadas marcaron un hito fundamental en el proceso de verificación del software para “*Easy Credit*”. Al adoptar esta estrategia, experimentamos los siguientes beneficios:

*Eficiencia Mejorada:* La automatización de pruebas permitió la ejecución rápida y repetitiva de escenarios críticos, acelerando significativamente el proceso de verificación. Esto



resultó en una identificación ágil y temprana de problemas potenciales, facilitando su resolución antes de que impactaran negativamente en la calidad del software.

*Detección de Problemas no Evidentes:* Las pruebas automatizadas no solo confirmaron la funcionalidad esperada, sino que también revelaron problemas no evidentes durante el desarrollo manual. Esto mejoró la calidad del código y fortaleció la confianza en la robustez del sistema.

### ***11.2.2 Alta Participación en Pruebas de Seguridad:***

La colaboración efectiva en las pruebas de seguridad fue un elemento clave para garantizar la integridad y robustez de “Easy Credit”. Esta participación exitosa incluyó:

*Identificación Temprana de Vulnerabilidades:* La ejecución de pruebas de seguridad de manera integral y coordinada permitió abordar estos problemas antes de que pudieran ser explotados en un entorno de producción.

*Aplicación Más Robusta:* La participación activa en pruebas de seguridad contribuyó a una aplicación más robusta y resistente a posibles amenazas. La identificación y mitigación proactiva de riesgos mejoró la seguridad general del sistema.

### ***11.2.3 Integración con Otras Pruebas:***

La coordinación efectiva permitió una integración armoniosa de las pruebas de seguridad con otras fases de verificación, como pruebas de carga, rendimiento, funcionalidad, usabilidad y las pruebas unitarias realizadas durante la fase de desarrollo.

### 11.3 Desafíos Encarados:

#### 11.3.1 *Desafíos en la Integración de Componentes:*

Durante el proceso de verificación, nos encontramos con desafíos significativos al integrar diversos componentes de “*Easy Credit*”. Estos desafíos incluyeron:

*Interdependencia Compleja:* La interdependencia compleja entre los distintos módulos del software generó obstáculos en la integración. Las conexiones no previstas y los efectos secundarios inesperados surgieron, lo que requirió un análisis profundo y ajustes cuidadosos, que aún se encuentran en desarrollo, al ser un proyecto de carácter formativo y que depende en gran manera de la curva de aprendizaje.

*Problemas de Comunicación:* La falta de una comunicación clara y continua entre el equipo inicial y la deserción estudiantil de la mayoría de ellos, recargo la responsabilidad de diferentes componentes a una sola persona, lo que contribuyó a la dificultad en la resolución rápida de problemas de integración. Esto llevó a retrasos temporales en el proceso de verificación.

*Ajustes en la Arquitectura:* Ha sido necesario realizar ajustes en la arquitectura para facilitar una integración más suave. La reestructuración de ciertos componentes y la implementación de interfaces más claras se volvieron imperativas para superar estos desafíos, que aún se encuentran en desarrollo.

### ***11.3.2 Dificultades en la Planificación de Pruebas:***

La planificación de las pruebas se presentó como otro desafío clave en el proceso de verificación de “*Easy Credit*”. Estas dificultades incluyeron:

*Complejidad del Proyecto:* La complejidad inherente al proyecto “*Easy Credit*”, pese a ser una solución pequeña, abarca varios módulos y múltiples funcionalidades, para ser desarrollada por una sola persona, lo que dificultó la creación de un plan de pruebas completo desde el principio. La adaptabilidad y la flexibilidad se volvieron esenciales.

*Falta de Experiencia en Algunos Aspectos:* La falta de experiencia en ciertos aspectos, como la gestión de grandes volúmenes de datos o la interacción con sistemas externos, uso e instalación de aplicaciones en versiones no compatibles, Frameworks, entre otras, generó incertidumbre en la planificación. Aprender sobre la marcha resultó inevitable.

*Alineación con Plazos Formativos:* Dado que “*Easy Credit*” forma parte de un proceso formativo con plazos predefinidos, la planificación de pruebas se vio influida por la necesidad de cumplir con estos plazos, lo que generó tensiones adicionales.

## **11.4 Lecciones Aprendidas:**

### ***11.4.1 Importancia de la Comunicación:***

La experiencia en “*Easy Credit*” resaltó la importancia crítica de una comunicación efectiva en todos los niveles del equipo. Aprendimos que:

Transparencia Promueve Soluciones Rápidas: Mantener líneas abiertas y transparentes de comunicación facilita la identificación temprana de problemas, permitiendo soluciones más rápidas y efectivas.

Coordinación entre Equipos: La coordinación efectiva entre equipos, especialmente en proyectos complejos con múltiples módulos o funcionalidades, es esencial para evitar malentendidos y garantizar una integración suave.

Retroalimentación Constructiva: Fomentar un ambiente donde la retroalimentación constructiva sea bienvenida mejora la calidad del proceso de verificación. Se reconoce que los errores son oportunidades de mejora.

#### **11.4.2 Planificación Realista:**

Las lecciones aprendidas en la planificación de “*Easy Credit*” subrayaron la importancia de:

Evaluación Continua de la Complejidad: La evaluación continua de la complejidad del proyecto permite una planificación realista. Pues al ser una solución inventada sin ningún tipo de asesoramiento, como sucedió al inicio de la formación académica; no permitió reconocer la magnitud de los desafíos desde el principio, lo que es necesario para una distribución más precisa de los recursos.

*Flexibilidad en la Planificación:* La flexibilidad en la planificación es esencial en proyectos dinámicos. Adaptarse a cambios y ajustar estrategias según las lecciones aprendidas durante el desarrollo mejora la eficacia general del proceso.

#### ***11.4.3 Calidad en las Pruebas Unitarias:***

La realización de pruebas unitarias en “Easy Credit” generó lecciones valiosas:

*Enfoque en la Base del Código:* Las pruebas unitarias constituyen la base del código. Invertir tiempo en pruebas unitarias sólidas reduce la probabilidad de errores en etapas posteriores y facilita la identificación temprana de problemas.

*Mantenimiento de la Integridad del Código:* Las pruebas unitarias no solo evalúan la funcionalidad, sino que también contribuyen a mantener la integridad del código. La implementación constante de pruebas unitarias mejora la confianza en el sistema.

### **11.5 Recomendaciones para Mejoras Futuras:**

#### ***11.5.1 Mayor Enfoque en Pruebas Unitarias:***

Recomendamos un mayor énfasis en las pruebas unitarias para fortalecer la base del código. Este enfoque asegurará una mayor robustez del sistema, reduciendo la posibilidad de errores y facilitando la identificación temprana de problemas.

### ***11.5.2 Incorporación Temprana de Pruebas de Rendimiento:***

Sugerimos la inclusión de pruebas de rendimiento en las etapas iniciales del desarrollo. Esto permitirá identificar y abordar problemas de rendimiento desde el principio, garantizando una experiencia fluida y eficiente para los usuarios finales.

### ***11.5.3 Mejora en la Planificación de Pruebas:***

Para proyectos futuros, se recomienda una planificación más detallada y adaptable. Evaluar la complejidad y la experiencia requerida en etapas tempranas contribuirá a una planificación más realista y a la adaptación a posibles cambios en el desarrollo.

### ***11.5.4 Feedback Continuo y Evaluación de Desempeño:***

Fomentar un ciclo de retroalimentación continuo y una evaluación de desempeño periódica ayudará a mejorar la eficiencia del equipo y a identificar oportunidades de mejora. Esto garantizará un proceso de verificación más ágil y eficaz.

### ***11.5.5 Documentación Clara y Accesible:***

La documentación del software, incluyendo manuales técnicos y de usuario, debe ser clara, detallada y accesible. Esto facilitará la comprensión del sistema y reducirá el tiempo dedicado a resolver problemas relacionados con la falta de información.

A continuación, se presentarán las bitácoras de los procesos documentales que permitieron registrar los avances de “*Easy Credit*”, resultados preliminares, obstáculos, ideas y datos.

### 11.5.5.1 Bitácoras de algunos procesos documentados:

Tabla 11. Bitácora documentación de Pruebas

Documentación de Pruebas		Fecha: 11/12/2023
<b>Avance:</b>	<i>Se crearon casos de prueba detallados, especificando los resultados esperados y los procedimientos de prueba</i>	
<b>Obstáculos:</b>	<i>Desafíos al simular ciertos escenarios de prueba realistas, ajustes necesarios</i>	
<b>Buenas Prácticas:</b>	<i>Se adoptó una metodología de pruebas centrada en el usuario para asegurar una cobertura exhaustiva</i>	
<b>Datos:</b>	<i>Registro de resultados de pruebas y correcciones realizadas durante las pruebas</i>	
<b>Soporte:</b>	<i>Documento de Pruebas detallando casos de prueba. Registro de resultados y correcciones</i>	

Tabla 12. Bitácora, manuales de Usuario

Manuales de Usuario		Fecha: 05/02/2024
<b>Avance:</b>	<i>Se elaboró un manual de usuario completo, proporcionando instrucciones claras para los usuarios finales</i>	
<b>Obstáculos:</b>	<i>Interpretación variada de algunas secciones por parte de los usuarios, de acuerdo a los roles requiere clarificación</i>	
<b>Buenas Prácticas:</b>	<i>Se incluyeron capturas de pantalla y ejemplos para mejorar la comprensión del usuario</i>	
<b>Datos:</b>	<i>Recopilación de comentarios de usuarios durante la revisión del manual</i>	
<b>Soporte:</b>	<i>Manual de Usuario con capturas de pantalla</i>	

Tabla 13. Bitácora documentación de configuración

Documentación de Configuración		Fecha: 11/02/2024
<b>Avance:</b>	<i>Se documentaron las configuraciones del entorno de desarrollo, dependencias y configuraciones</i>	
<b>Obstáculos:</b>	<i>Cambios en las dependencias externas afectaron la configuración, actualizaciones necesarias</i>	
<b>Buenas Prácticas:</b>	<i>Se utilizó un archivo de configuración centralizado y se implementó una gestión eficiente de dependencias</i>	
<b>Datos:</b>	<i>Registro de cambios en la configuración y su impacto en el desarrollo</i>	
<b>Soporte:</b>	<i>Documento de Configuración del Entorno con registro de cambios y gestión de dependencias</i>	

Tabla 14. bitácora documentación de despliegue

Documentación de Despliegue		Fecha: 10/02/2024
<b>Avance:</b>	<i>Se detallaron los pasos para el despliegue de la aplicación en un entorno de producción y de prueba</i>	
<b>Obstáculos:</b>	<i>Problemas de compatibilidad con la infraestructura existente, adaptaciones necesarias</i>	
<b>Buenas Prácticas:</b>	<i>Se automatizó parte del proceso de despliegue para garantizar consistencia y reducir posibles errores humanos</i>	
<b>Datos:</b>	<i>Registro de problemas y soluciones durante los despliegues anteriores</i>	
<b>Soporte:</b>	<i>Plan detallado de Mantenimiento y Soporte con procedimientos detallados</i>	

Tabla 15. Bitácora documentación de mantenimiento

Documentación de Mantenimiento		Fecha: 11/02/2024
<b>Avance:</b>	<i>Se documentaron aspectos críticos del código, dependencias y decisiones de diseño para facilitar futuras actualizaciones y mantenimiento</i>	
<b>Obstáculos:</b>	<i>Algunos cambios en los requisitos, proceso de manejo de cambios ajustado</i>	
<b>Buenas Prácticas:</b>	<i>Se aplicaron comentarios detallados en áreas propensas a cambios para guiar a los desarrolladores en futuras modificaciones</i>	
<b>Datos:</b>	<i>Registro de actualizaciones y cambios realizados durante el mantenimiento</i>	
<b>Soporte:</b>	<i>Documento de Mantenimiento detallando aspectos críticos y registro de cambios</i>	

Tabla 16. Bitácora, Manual Técnico

Manual Técnico		Fecha: 16/02/2024
<b>Avance:</b>	<i>Se realizó la redacción del manual técnico, detallando la arquitectura del sistema, tecnologías utilizadas y procesos de desarrollo</i>	
<b>Obstáculos:</b>	<i>Identificación de secciones técnicas complejas, requirieron simplificación</i>	
<b>Buenas Prácticas:</b>	<i>Se estructuró el manual de manera clara y se incluyeron diagramas para facilitar la comprensión técnica</i>	
<b>Soporte:</b>	<i>Manual Técnico con estructura de acuerdo al material de estudio</i>	



Tabla 17. Bitácora Plan de Mantenimiento y Soporte

Plan de Mantenimiento y Soporte		Fecha: 18/02/2024
<b>Avance:</b>	<i>Se elaboró un plan detallado para el mantenimiento continuo y el soporte técnico, incluyendo la asignación de roles y responsabilidades</i>	
<b>Obstáculos:</b>	<i>Expectativas divergentes sobre el alcance del soporte, requiere alineación</i>	
<b>Buenas Prácticas:</b>	<i>Se establecieron procedimientos claros para la gestión de problemas y la aplicación de actualizaciones</i>	
<b>Soporte:</b>	<i>Plan detallado de Mantenimiento y Soporte con procedimientos detallados</i>	

Tabla 18. Bitácora Plan de Capacitación

Plan de Capacitación		Fecha: 18/02/2024
<b>Avance:</b>	<i>Se elaboró el plan de capacitación, identificando las necesidades de formación para el equipo de desarrollo y usuarios finales</i>	
<b>Obstáculos:</b>	<i>Variación en las necesidades de capacitación entre los miembros del equipo o usuarios finales, ajustes necesarios</i>	
<b>Buenas Prácticas:</b>	<i>Se diseñó el plan teniendo en cuenta diferentes estilos de aprendizaje y niveles de experiencia</i>	
<b>Soporte:</b>	<i>Plan de Capacitación con secciones sobre estilos de aprendizaje y necesidades de formación</i>	

Tabla 19. Bitácora Plan de Migración y Respaldo de Datos

Plan de Migración y Respaldo de Datos		Fecha: 19/02/2024
<b>Avance:</b>	<i>Se trabajó en el plan de migración de datos y en la estrategia de respaldo para garantizar la integridad y disponibilidad de la información</i>	
<b>Obstáculos:</b>	<i>Problemas de compatibilidad con algunos sistemas de respaldo, investigación en curso</i>	
<b>Buenas Prácticas:</b>	<i>Se realizaron pruebas de migración simuladas y se documentaron los procedimientos de respaldo de manera detallada</i>	
<b>Soporte:</b>	<i>Estrategia de respaldo y migración con detalles y resultados de pruebas simuladas</i>	

## Conclusiones

Realizar una exhaustiva evaluación del producto de software “*Easy Credit*” y llevar a cabo una serie de ajustes en los procesos de desarrollo, ha sido fundamental para abordar los desafíos enfrentados durante todas las etapas para su desarrollo. Siendo un proyecto educativo llevado a cabo por un solo aprendiz, se encontraron limitaciones en términos de recursos humanos y experiencia; sin embargo, mediante la identificación de áreas de mejora y la implementación de cambios estratégicos, como la introducción de pruebas unitarias automáticas, revisiones continuas de código y optimización de procesos, se logró mejorar la eficiencia y la calidad del desarrollo de software.

La implementación de prácticas de calidad, la selección adecuada de metodologías y herramientas, y la aplicación de acciones correctivas y preventivas han permitido abordar los desafíos identificados y fortalecer la solidez del software. En resumen, este proceso de evaluación y mejora ha sido un viaje de aprendizaje invaluable que ha permitido no solo superar los obstáculos, sino también sentar las bases para un desarrollo más efectivo y un aprendizaje continuo en futuros proyectos. La experiencia obtenida en el proceso de desarrollo de la solución de software propuesta para el proceso formativo, servirá como una valiosa lección para abordar de manera más efectiva los desafíos futuros y garantizar el éxito en proyectos similares.

## Referencias

*Aplicación de pruebas de software.* (s.f.) Recuperado de:

<https://sena.territorio.la/content/index.php/institucion/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF48/index.html#/curso/tema1>

OpenAI. (2024). ChatGPT. (Versión 3.5 de enero de 2022). [Modelo de lenguaje de gran tamaño].

<https://chat.openai.com/chat>