	INFORME FINAL DE AUDITORÍA INTERNA DEL PROYECTO DE GESTIÓN DE PAGOS EN URBANIZACIÓN MANANTIAL	DCCO(Departamento de Ciencias de la Computación)	
		N.º Informe	GPUM-2025-001
		Página:	1 de 3

A.BASE LEGAL

El presente informe se basa en las actividades realizadas durante la auditoría interna llevada a cabo desde el 11 de diciembre de 2024 hasta el 19 de enero del 2025, conforme a las normas ISO 9000 y ISO/IEC/IEEE 29119-5:2016. Estas normas proporcionaron el marco teórico y práctico necesario para garantizar la correcta ejecución de los procesos auditados.

B. ANTECEDENTES

El Durante la auditoría, se realizaron las siguientes actividades:

1. Presentación inicial del equipo auditor y auditado.
2. Exposición de la agenda y los objetivos de la auditoría.
3. Simulación de los escenarios prácticos y resolución de inconvenientes.
4. Compromiso de realizar una revisión futura del mapa de procesos.
5. Revisión del backlog, sprints, requisitos funcionales y casos de prueba relacionados con el proyecto “Gestión de pagos de la urbanización Manantial”.
6. Implementación de pruebas de aceptación mediante la herramienta Cucumber.
7. Evaluación de la gestión del proyecto en JIRA.
8. Resolución de no conformidades menores detectadas.


C. OBJETIVO

Verificar el cumplimiento de los estándares internacionales de calidad ISO 9000 e ISO/IEEE 291195:2016 en los procesos del proyecto auditado, identificando fortalezas, áreas de mejora y la efectividad de las acciones correctivas implementadas.

D. DESARROLLO

A. Metodología

1. Realizamos una reunión de apertura donde se dio a conocer la documentación, el funcionamiento del proyecto junto con su documentación (mapa de 14 niveles) y la presentación del plan de pruebas con su respectivo cronograma.
2. Análisis de la documentación entregada por el equipo, que incluye el plan de pruebas, cronograma gestionado en JIRA, reportes de Cucumber y el informe de escenarios.


	INFORME FINAL DE AUDITORÍA INTERNA DEL PROYECTO DE GESTIÓN DE PAGOS EN URBANIZACIÓN MANANTIAL	DCCO(Departamento de Ciencias de la Computación)	
		N.º Informe	GPUM-2025-001
		Página:	2 de 3

3. Reunión a través de Google Meet con los responsables del equipo, incluyendo al líder del equipo y el personal auditado.
4. Evaluación de la implementación de las pruebas de aceptación realizadas con Cucumber en función de los requisitos funcionales establecidos.


B. Hallazgos 1. Documentación

a.En el plan que se realizó se cumple 28 cláusulas de la norma en las que se va detallando a continuación:

1. **Cláusula 2.3. Tailored conformance (Conformidad adaptada):** Explica cómo adaptar partes de la norma para proyectos específicos, siempre que las razones estén documentadas.
2. **Cláusula 4. Terms and definitions (Términos y definiciones):** Se define terminos de las pruebas que se basan en palabras claves.
3. **Cláusula 5.1. Overview (Visión general):** Introduce el concepto de pruebas modulares utilizando palabras clave para facilitar la creación, mantenimiento y reutilización de casos de prueba.Cláusula 7.3.7: Uso del motor de ejecución de pruebas automatizadas.
4. **Cláusula 5.2.1. Overview (Visión general):** En las pruebas basadas en palabras clave se representadas en diferentes niveles de abstracción. El cual , se logran organizar en en una o mas capas.
5. **Cláusula 5.2.2. Domain layer (Capa de dominio):** Enfocada en actividades del negocio, con palabras clave alineadas a los procesos funcionales.
6. **Cláusula 5.2.4. Multiple layers (Capas múltiples):** Para que se pueda combinar multiples capas es requerido un framework para gestionar las palabras clave
7. **Cláusula 5.3.1. Simple keywords (Palabras clave simples):** Se refiere a las acciones basicas que sean utilizadas en la capa de la interfaz puede permitir la conexion con herramientas de ejecucion de pruebas y palabras clave..
8. **Cláusula 5.3.3. Navigation/interaction (input) and verification (output):** Describe palabras clave utilizadas para entradas (navegación) y verificaciones de salida.
9. **Cláusula 5.3.4. Keywords and test result (Palabras clave y resultados de prueba):** El uso de palabras clave para llegar al estado de pruebas y capturar los resultados obtenidos.

	INFORME FINAL DE AUDITORÍA INTERNA DEL PROYECTO DE GESTIÓN DE PAGOS EN URBANIZACIÓN MANANTIAL	DCCO(Departamento de Ciencias de la Computación)	
		N.º Informe	GPUM-2025-001
		Página:	3 de 3

10. **Cláusula 5.4. Keywords and Data (Palabras clave y datos):** Las palabras clave que se asocien con datos , el cual , necesitaran parametros. Esto es una mejora a las pruebas de palabras clave.
11. **Cláusula 6.1. Overview (Visión general):** Varios conceptos que colaborar con implementar pruebas de palabras clave exitosamente.No todos los conceptos son necesarios pero puede dar una mejora al diseño de las pruebas.
12. **Cláusula 6.2. Identifying keywords (Identificación de palabras clave):** Definir palabras clave en base a conjunto de acciones que sean frecuentes en las pruebas y se puede llegar a aplicar en diferentes circunstancias o situaciones.
13. **Cláusula 6.3. Composing test cases (Composición de casos de prueba):**Los casos de pruebas se van definiendo conforme se determinaron las palabras claves el cual tambien llega a la posibilidad que despues de hacer este proceso , existan más palabras claves.
14. **Cláusula 6.4. Keywords and data-driven testing (Pruebas basadas en palabras clave y datos):** Para tener una mejor calidad de software, se puede realizar una combinacion de parametros el objetivo es realizar varias pruebas con diferentes datos.
15. **Cláusula 6.5. Modularity and refactoring (Modularidad y refactorización):** Creacion de diferentes parametros para non tener problemas en el mantenimiento cuando se creen nuevas pruebas o nuevas personas.Para ello, se crea un rastreo de palabras claves que sean utilizados con frecuencia ademas,requerir de una persona responsable que tenga todas esas palabras claves, agregar y actualizar, tener reuniones en ciertos tiempos para revisar las palabras clave y el cambio de estas palabras este limitado y si se cambia debe estar muy bien documentado.
16. **Cláusula 6.6.1 TD1 Identify Feature Sets:** Identificación de funcionalidades clave.
17. **Cláusula 6.6.2 TD2 Derive Test Conditions:** Definición de condiciones para evaluar requisitos.
18. **Cláusula 6.6.3 TD3 Derive Test Coverage Items:** Garantiza que los casos de prueba cubran todos los requisitos.
19. **Cláusula 6.6.4 TD4 Derive Test Cases:** Desarrollo de casos de prueba específicos.
20. **Cláusula 6.6.5 TD5 Assemble Test Sets:** Agrupación lógica de casos de prueba.

	INFORME FINAL DE AUDITORÍA INTERNA DEL PROYECTO DE GESTIÓN DE PAGOS EN URBANIZACIÓN MANANTIAL	DCCO(Departamento de Ciencias de la Computación)	
		N.º Informe	GPUM-2025-001
		Página:	4 de 3


21. **Cláusula 7.1. Overview:** Explica los fundamentos para implementar un marco de trabajo efectivo.
22. **Cláusula 7.2. Components of a Keyword-Driven Testing framework:** Se describe los elementos que se componen en un marco que se base en palabras clave, eso incluye herramientas personalizadas o solución en forma de bibliotecas de scripts
23. **Cláusula 7.2.5 Asistente de prueba manual:** El asistente de prueba manual solo es necesario para la ejecución de pruebas manuales.
24. **Cláusula 7.2.7 Entorno de ejecución de pruebas y motor de ejecución:** Para respaldar las pruebas automatizadas basadas en palabras clave, el entorno de pruebas contendrá un motor de ejecución con enlaces al elemento en prueba.
25. **Cláusula 7.2.8 Biblioteca de palabras clave:** La biblioteca de palabras clave almacena definiciones de palabras clave para uno o más proyectos o partes de esos proyectos.
26. **Cláusula 7.2.9 Datos:** Utilizar datos que ayuden a los casos de pruebas que se basen en las palabras clave.
27. **Cláusula 7.3. Basic attributes of the Keyword-Driven Testing framework:** Se define los atributos que son necesarios para aplicar pruebas en base a las palabras clave.
28. **Cláusula 7.3.7 Motor de ejecución de pruebas:** Los motores de ejecución de pruebas están diseñados para ejecutar casos de prueba al abordar una o más interfaces de prueba (por ejemplo, una API, una GUI o una interfaz de hardware).

2. Ejecución de los casos de pruebas

- a. Los Escenarios : crear eventos , crear usuarios , registro de cobros y login son casos los casos de prueba para ejecutar en cucumber.
- b. Se crearon 2 escenarios de cada caso de prueba, dando un total de 8 escenarios.
- c. De los 8 escenarios preparados:
 1. 4 pasaron exitosamente bajo condiciones correctas.
 2. 4 fallaron debido a valores no admisibles, validando el comportamiento esperado del sistema.

3. Gestión del proyecto con JIRA Software

- a. JIRA se utilizó para visualizar y gestionar las actividades de pruebas de aceptación.

	INFORME FINAL DE AUDITORÍA INTERNA DEL PROYECTO DE GESTIÓN DE PAGOS EN URBANIZACIÓN MANANTIAL	DCCO(Departamento de Ciencias de la Computación)	
		N.º Informe	GPUM-2025-001
		Página:	5 de 3

- b. Permitió la correcta trazabilidad de tareas y reportes relacionados con los avances del proyecto y del plan de pruebas.

4. Resultados obtenidos en las pruebas

- Los escenarios exitosos demostraron que el sistema funciona correctamente bajo condiciones previstas.
- Los escenarios fallidos fueron diseñados para validar errores y comprobaciones de seguridad.

5. No conformidades


- Se presentó una no conformidad a lo largo del cronograma y las reuniones con el auditor.
- La no conformidad fue solucionada en base a las directrices proporcionadas por el auditor
- La documentación fue actualizada para cumplir con las cláusulas de la norma ISO/IEC/IEEE 29119-5:2016.

E. CONCLUSIONES

- El proyecto cumple con los requisitos establecidos en las normas ISO/IEC/IEEE 291195:2016 e ISO 9000:2005.
- El plan de auditorio conjunto con las cláusulas mejora el proceso de auditoria y aseguran la documentación adecuada para futuras auditorias y revisiones.

F. RECOMENDACIONES

- Implementar un programa de auditorías periódicas para asegurar la mejora continua de los procesos y de la calidad de software.
- Promover el uso de herramientas colaborativas para mejorar la gestión de proyectos y pruebas.
- Tener en cuenta un especialista auditor es eficiente en los programas, se puede evitar varios errores de calidad e identificar mejoras según el estándar seleccionado.
- Tener en cuenta los estándares o normas es eficiente en el proceso de análisis y diseño de software, la documentación va a ir de acuerdo con un estándar y cumplirá con el aseguramiento de software. Además, dentro del ciclo de vida del software se evitará varias inconsistencias, errores y no conformidades.




	INFORME FINAL DE AUDITORÍA INTERNA DEL PROYECTO DE GESTIÓN DE PAGOS EN URBANIZACIÓN MANANTIAL	DCCO(Departamento de Ciencias de la Computación)	
		N.º Informe	GPUM-2025-001
		Página:	6 de 3

Sangolquí, 12 de febrero de 2025

G. REFERENCIAS

- a. International Organization for Standardization. (2016). ISO/IEC/IEEE 29119-5:2016 – Software and systems engineering – Software testing.

H. APROBACIÓN

Rubro	Nombre Apellido	Unidad /Cargo	Firma
Elaborado por	Francisco Terán	Líder de Equipo	
Elaborado por	Javier Gonzaga	Auditado	
Elaborado por	Richard Gualotuña	Auditado	
Aprobado por			