	INFORME DE HECHOS LA REUNIÓN DE SEGUIMIENTO DE LA AUDITORÍA INTERNA	Departamento de Ciencias de la Computación	
		N.º Informe	DCCO-2025-001
		Página:	1 de 2

A. BASE LEGAL

La reunión de auditoría se realizó el día 10 de enero del año 2025, en ella se estableció el marco de trabajo fundamentado en la norma ISO-IEC-IEEE-29119-5 y se realizó una descripción general sobre el backlog, sprints, requisitos funcionales, casos de prueba, implementación de las pruebas mediante Cucumber, además de evaluar los resultados obtenidos de los escenarios planteados de prueba del proyecto “Creación de grupos de investigación dentro de la universidad”.

B. ANTECEDENTES

1. Se realizó la presentación del equipo y una reunión inicial entre los integrantes, permitiendo entender el contexto y objetivo del proyecto, estableciendo el marco de trabajo, las líneas de comunicación, las normas y directrices a utilizar, marcando mejoras a implementar para las siguientes reuniones.
2. Se desarrolló la reunión en base al itinerario establecido con anterior en el grupo, en base a disponibilidad entre los integrantes y el auditor, tratando la auditoría en base a los puntos establecidos en el plan de auditoría.
3. El equipo auditado utilizó Jira como herramienta para gestionar el proyecto
4. El desarrollo de los casos de prueba se realizó en base a la norma ISO/IEC/IEEE 29119-5:2026
5. Se hizo uso de Cucumber para la integración de los casos de prueba en Jira


C. OBJETIVO

El objetivo de la reunión fue realizar la revisión del backlog, requisitos funcionales, casos de prueba e implementación de los casos de prueba mediante el Cucumber con los resultados de cada escenario. Todo esto debe regirse por la norma ISO/IEC/IEEE 29119-5:2016.

D. DESARROLLO

A. Metodología

1. Se inicio con el saludo entre el auditor y los integrantes del equipo de trabajo.
2. Se comprobó la asistencia de todos los integrantes
3. Se revisó el itinerario y las actividades a realizar.
 - a. Listado de asistencia
 - b. Revisión de las clausulas de la norma ISO 29119-5:2016
 - c. Presentación del Backlog del proyecto
 - d. Revisión del documento de los casos de prueba
 - e. Explicación de los objetivos de cada caso de prueba
 - f. Explicación de los requisitos funcionales.
 - g. Explicación de los escenarios de prueba.
 - h. Revisión de reportes generados y evidencias de las pruebas realizadas mediante Cucumber.
 - i. Compromisos y recomendaciones finales.
4. Se realizó la revisión del plan de auditoría del proyecto
5. Se revisó la especificación de todas las clausulas de la norma ISO 29119-5:2016y el uso de las mismas en el proyecto
6. Se realizó la revisión del Backlog con enfoque a las actividades de la integración de las pruebas
7. Se realizó la revisión de los casos de prueba de forma manual para concordancia con lo establecido en Jira
8. Se realizó la revisión de los requisitos funcionales.

	INFORME DE HECHOS LA REUNIÓN DE SEGUIMIENTO DE LA AUDITORÍA INTERNA	Departamento de Ciencias de la Computación	
		N.º Informe	DCCO-2025-001
		Página:	2 de 2


9. Se explicaron y revisaron los casos de prueba en los escenarios establecidos en Cucumber
10. Se revisaron los resultados obtenidos de las pruebas mediante Cucumber.
11. Se designaron compromisos para la próxima reunión.

B. Hallazgos

1. Documentación


El plan de pruebas elaborado por el equipo se rige por la norma ISO 29119-5:2016, realizando el uso de las siguientes cláusulas:

- a. 6.2 Identificación de palabras clave Es una tarea fundamental en las pruebas basadas en palabras clave, ya que el contenido, la granularidad y la estructura de las palabras clave pueden afectar la forma en que se definen los casos de prueba de palabras clave. Es importante nombrar las palabras clave de una manera que parezca natural para las personas que trabajarán con ellas.
- b. 6.3 Redacción de casos de prueba La creación de casos de prueba basados en palabras clave se realiza utilizando palabras clave previamente definidas, las cuales pueden documentarse en tablas o bases de datos. Por lo general, estos casos emplean palabras clave de una sola capa, lo que facilita que diferentes evaluadores trabajen en el diseño de distintas capas de manera independiente.
- c. 6.4 Palabras clave y pruebas basadas en datos Las pruebas que combinan el uso de datos y palabras clave, con parámetros separados, posibilitan la reutilización de secuencias de acciones utilizando diferentes conjuntos de datos. Estos datos, almacenados en tablas o bases de datos, permiten generar nuevas pruebas de manera eficiente.
- d. 6.5 Modularidad y refactorización. La modularidad en las pruebas basadas en palabras clave mejora la longevidad de los casos de prueba, pero puede generar problemas de mantenimiento, como palabras clave redundantes y sin uso, y conflictos por cambios descoordinados. Para mitigarlos, considere hacer referencias cruzadas de las palabras clave utilizadas, designar una autoridad responsable, realizar revisiones periódicas de las palabras clave y documentarlas claramente por capa y uso.
- e. 6.6 Pruebas basadas en palabras clave en el proceso de diseño de pruebas La norma ISO/IEC/IEEE 29119-5 describe un proceso crítico para el diseño e implementación de pruebas, particularmente relevante para las pruebas basadas en palabras clave. Este enfoque se centra en la creación de casos de prueba utilizando palabras clave, estableciendo condiciones previas, seleccionando entradas, definiendo acciones y verificando resultados. Los conjuntos de pruebas se ensamblan a partir de estos casos, empleando palabras clave claras para garantizar una ejecución eficaz y condiciones de prueba adecuadas.
- f. 6.7 Convertir casos de prueba que no están basados en palabras clave en pruebas basadas en palabras clave. Introducir pruebas basadas en palabras clave en un proyecto existente implica convertir los casos de prueba actuales en casos basados en palabras clave. Los beneficios incluyen uniformidad en la estructura para una mejor legibilidad y mantenimiento, posible reutilización de palabras clave identificadas, uso consistente del marco de automatización y accesibilidad para evaluadores no técnicos. Sin embargo, puede ser preferible mantener los casos

	INFORME DE HECHOS LA REUNIÓN DE SEGUIMIENTO DE LA AUDITORÍA INTERNA		Departamento de Ciencias de la Computación
			N.º Informe DCCO-2025-001
			Página: 3 de 2

existentes debido a su volumen, automatización establecida y rentabilidad en comparación con la conversión.


- g. 5.1 "Keyword-Driven Testing is an approach that uses keywords to define test cases."
- h. 5.2 "Keywords can be organized in multiple layers to enhance test case creation and execution."
- i. 5.3 "Simple keywords are straightforward actions, while composite keywords combine multiple actions."
- j. 5.3.3 "Keywords can be categorized into navigation steps and verification steps."
- k. 5.3.4 "Keywords can determine test outcomes based on their execution results."
- l. 5.4 "Keywords can be enhanced by associating them with data for better management."
- m. 7.1 If Keyword-Driven Testing is to be applied, everything necessary to do that needs to be organized in a Keyword-Driven Testing framework. This framework consists of concepts, documents and tools.
- n. 7.2.1 Keyword-Driven Testing is typically supported by a framework. The framework can be realized in a variety of ways including by commercial tools, custom tools, and solutions in the form of script libraries or other supporting elements.
- o. 7.2.3 Decomposer The decomposer is required if composite keywords are used. The main task of the decomposer is to transform the keyword test case, which consists of a sequence of high-level keywords, into the appropriate sequence of low-level keywords.
- p. 7.2.4 Data sequencer The data sequencer is required if Keyword-Driven Testing is to be applied with several sets of data associated with one keyword test case. The main task of the data sequencer is to transform the sequence of keywords (e.g. low-level or high-level) which are not yet associated with data to a list of keywords with specific data.
- q. 7.2.6 Tool bridge The tool bridge is only required for automated test execution and has a similar function to the manual test assistant used to support manual test execution.
- r. 7.2.7 Test execution environment and execution engine To support automated Keyword-Driven Testing, the test environment will contain an execution engine with links to the item under test. The execution engine is a tool implemented either by software, hardware or both. Its task is to execute the test cases by performing the actions associated with the keywords.
- s. 7.2.8 Keyword library The keyword library stores keyword definitions for one or more projects or portions of those projects. It is used to store the core information on keywords, such as: name, description, parameters and, in the case of composite keywords, the list of keywords from which the respective keyword is composed or derived. For test automation, it also contains necessary information for the tool bridge to associate the Keywords with the keyword execution code. The keyword library can help the tester to find a keyword.
- t. 7.2.10 Script repository The Script repository stores keyword execution code. It is only required if Keyword-Driven Testing is done with the aim of executing the test cases automatically.

	INFORME DE HECHOS LA REUNIÓN DE SEGUIMIENTO DE LA AUDITORÍA INTERNA		Departamento de Ciencias de la Computación	
			N.º Informe	DCCO-2025-001
			Página:	4 de 2

- u. 7.3.1 General information on basic attributes This clause defines framework attributes that are generally necessary for the application of Keyword-Driven Testing. It describes attributes which are required in Keyword-Driven Testing frameworks and are necessary for compliance with this International Standard.
- v. 7.3.2 General attributes: a) There shall be documentation recorded describing each keyword. b) There shall be documentation recorded for the parameters of each keyword. c) A default value shall be documented for every parameter in case a value for a parameter is missed in the keyword test case definition. e) There shall be high-level documentation recorded describing how data is stored and referenced for data driven tests.
- w. 7.3.6 Tool bridge Requirements that apply to a tool bridge include the following: a) The tool bridge shall provide the test execution engine with the appropriate execution code to execute the test cases.
- x. 7.3.7 Test execution engine Test execution engines are designed to execute test cases by addressing one or more test interfaces (e.g. an API, a GUI or a hardware interface). A test execution engine can be implemented by software, by hardware or both. A common example of this class of tools is "JUnit".
- y. 7.3.8 Keyword library Requirements that apply to the keyword library include the following: a) The keyword library shall support the definition of keywords that includes the basic attributes of name, description and parameters.
- z. 7.3.9 Script repository Requirements that apply to the script repository include the following: a) The script repository shall support the storage of keyword execution code. b) The script repository shall support the inclusion of references to allow keyword execution code to be associated with its corresponding keyword in the keyword library.
- aa. 7.4.1 General information on advanced attributes This subclause defines additional attributes that are recommended to achieve the full benefits of Keyword Driven Testing. Basic Keyword-Driven Testing is possible without these attributes. This subclause does not identify requirements necessary for compliance with this International Standard.
- bb. 7.4.5 Manual test assistant Recommendations that apply to the manual test assistant include the following: a) The manual test assistant should provide the capability to attach screenshots or other outputs of the test item to the test log.
- cc. 7.4.6 Tool bridge No advanced attributes are defined for the tool bridge.
- dd. 7.3.9 Script repository Requirements that apply to the script repository include the following: a) The script repository shall support the storage of keyword execution code. b) The script repository shall support the inclusion of references to allow keyword execution code to be associated with its corresponding keyword in the keyword library.

2. Ejecución de las pruebas

- a. Los casos de prueba encontrados en Cucumber cumplen con los escenarios establecidos para una correcta ejecución del sistema, donde se abordaron con éxito los que se encontraron aunque faltaron escenarios por implementar

	INFORME DE HECHOS LA REUNIÓN DE SEGUIMIENTO DE LA AUDITORÍA INTERNA	Departamento de Ciencias de la Computación	
		N.º Informe	DCCO-2025-001
		Página:	5 de 2

- b. Los escenarios que fallaban se abordaron las correcciones establecidas para una mejor calidad de producto

3. Gestión del proyecto

- a. Se hizo uso de Jira para poder organizar al equipo de trabajo de manera exitosa

4. Resultados obtenidos de las pruebas

- a. El equipo preparó 8 casos de prueba donde solo se pudo observar 3 ejecutados correctamente, por lo que quedó pendiente agregar los escenarios faltantes

E. CONCLUSIONES

1. La reunión de auditoría permitió evaluar el progreso y cumplimiento de los objetivos del proyecto
2. Gracias al desarrollo de la auditoria se identificaron las fortalezas, avances y las áreas que requieren atención por parte del equipo, además de establecer compromisos de mejora en la ejecución del proyecto, permitiendo un mejor enfoque y estructura en el desarrollo del proyecto
3. La colaboración entre los integrantes y el cumplimiento de las recomendaciones realizadas permitirá garantizar mejores procesos y resultados en el futuro

F. RECOMENDACIONES

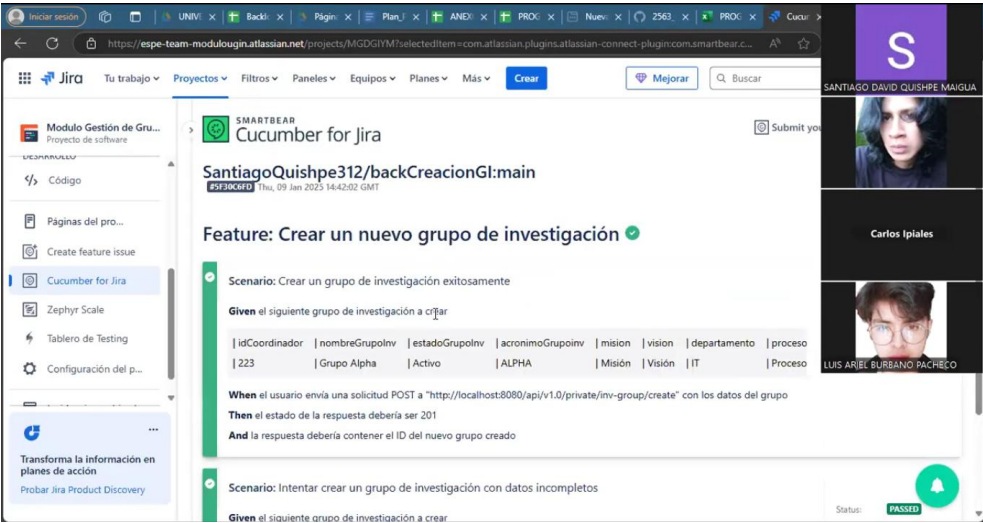
1. Mejorar los escenarios de pruebas para cubrir mayores casos, sobre todo en uso real de la aplicación, garantizando la correcta funcionalidad
2. Mantener una documentación clara, con evidencias de los procesos realizados, permitiendo un mejor análisis ya sea para auditoria o mejora de procesos.
3. Realizar auditorías periódicas que permitan validar las mejoras y una mejor implementación del estándar

Sangolquí, a 14 de enero de 2025

G. APROBACIÓN

Rubro	Nombre Apellido	Unidad /Cargo	Firma
Elaborado por	Luis Burbano	Estudiante	

H. ANEXOS



Anexo 1. Reunión con el grupo auditado

NO CONFORMIDAD N°1 (MENOR)	
Nº de NCM o nc	Descripción de la NC
NC1 Menor	<p>Se identificó que no se están integrando todas las pruebas en Jira, lo que limita la verificación de evidencias necesarias para validar las actividades realizadas por el equipo. Esta desviación afecta la trazabilidad y puede ocasionar problemas menores en la auditoría de seguimiento.</p> <p>Norma/Cláusula incumplida: ISO/IEC/IEEE 29119-5:2016 – Cláusula 7.2.5</p> <p>Evidencia Falta de reportes de los casos de prueba en Jira</p> <p>Impacto: Aunque no afecta gravemente a los objetivos de calidad, el que no estén presentes todas las evidencias necesarias presenta un incumplimiento de los procesos establecidos y dificulta la generación de informes y análisis, por lo que es necesario completar la integración con las demás pruebas</p> <p>Recomendación: Integrar las pruebas en Jira las pruebas faltantes Establecer un plan de acción para garantizar que las evidencias se registren en tiempo y forma adecuados. Realizar verificación de subida de las pruebas una vez realizada para comprobar que efectivamente están presentes</p> <p>Responsable: Líder del equipo</p> <p>Fecha Límite: 16/01/2025</p>

Anexo 2. No conformidades encontradas