

Curso: Aseguramiento de la Calidad de Software		Horas aula: 3
Clave: 061CE004		Horas plataforma: 1
Antecedentes: 061CP008		Horas laboratorio: 0
		Horas independientes: 2
Competencia del área: Implementar soluciones e innovaciones tecnológicas, con el fin de contribuir a una planeación responsable de los recursos humanos, tecnológicos y financieros, con base en las necesidades, la problemática analizada y los estándares de calidad establecidos por la organización.	Competencia del curso: Implementar programas que cumplan con la gestión de estrategias efectivas de Aseguramiento de Calidad del Software (SQA) a lo largo del ciclo de vida del sistema, mediante el uso y aplicación de herramientas SQA, con el fin de generar satisfacción en el cliente al identificar errores y defectos en etapas tempranas del desarrollo, aplicando las normas y estándares de calidad del desarrollo de software.	
Elementos de competencia:		
<ol style="list-style-type: none"> 1. Reconocer los conceptos esenciales del aseguramiento de calidad del software, conforme a estándares e instituciones pertinentes, para integrarlos en el proceso de desarrollo de sistemas de manera colaborativa y responsable, utilizando herramientas de análisis estático. 2. Comprender la estructura y los objetivos de los estándares y normas relacionados con la calidad del software, para liderar su implementación de manera responsable y efectiva en el proceso de desarrollo de sistemas, utilizando herramientas de gestión de pruebas. 3. Aplicar técnicas de calidad del software que permitan medir su eficacia mediante métricas adecuadas, para desarrollar programas computacionales fiables y eficientes que cumplan con los requerimientos del usuario dentro de las organizaciones, utilizando herramientas de análisis dinámico. 		
Perfil del docente:		
Licenciado en sistemas, ingeniero en software o a fin, preferente con Maestría en las áreas de Ciencias de la Computación, Ingeniería en Software o afín. El facilitador debe tener las competencias sobre los procesos de ingeniería en software y sus normas y estándares. Además deberá ser competente para desarrollar la función de la docencia que permita fungir como un facilitador en el proceso de enseñanza aprendizaje, atendiendo los lineamientos instituciones del modelo educativo vigente.		
Elaboró: MIGUEL ANGEL ROMERO OCHOA		Enero 2024
Revisó: GISELL FERNANDA GARCÍA ALBELAIS		Junio 2024
Última actualización:		
Autorizó: Coordinación de Procesos Educativos		

<p>Elemento de competencia 1: Reconocer los conceptos esenciales del aseguramiento de calidad del software, conforme a estándares e instituciones pertinentes, para integrarlos en el proceso de desarrollo de sistemas de manera colaborativa y responsable, utilizando herramientas de análisis estático.</p>	
<p>Competencias blandas a promover: Trabajo en equipo, comunicación efectiva, responsabilidad.</p>	
<p>EC1 Fase I: Fundamentos del aseguramiento de calidad del software</p>	
<p>Contenido: Conceptos esenciales del aseguramiento de calidad del software, estándares e instituciones pertinentes.</p>	
<p>EC1 F1 Actividad de aprendizaje 1: Presentación sobre conceptos esenciales del aseguramiento de calidad del software</p> <p>Preparar una presentación sobre los conceptos esenciales del aseguramiento de calidad del software, incluyendo definiciones, estándares relevantes e instituciones pertinentes. La presentación debe destacar la importancia de estos conceptos en el proceso de desarrollo de sistemas, explicando cómo se integran. Utilizar ejemplos y casos de estudio para ilustrar los puntos clave.</p> <p>2 hrs. Aula 2 hrs. Independientes</p>	<p>Tipo de actividad: Aula (X) Plataforma() Laboratorio () Grupal () Individual (X) Equipo () Independientes (X)</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Pressman, R. S. (2010). "Ingeniería del Software, Un enfoque práctico" (7ma. ed). McGraw-Hill. • IEEE Computer Society. (Última edición). Estándares de IEEE para el aseguramiento de calidad del software. <p>Criterios de evaluación de la actividad:</p> <p>Rúbrica de exposición .</p>
<p>EC1 F1 Actividad de aprendizaje 2: Cuadro comparativo sobre herramientas de análisis estático.</p> <p>Realizar un análisis comparativo de herramientas de análisis estático como SonarQube y Checkstyle. Investigar sus características, funcionalidades, ventajas y limitaciones. Identificar cómo estas herramientas pueden ser utilizadas de manera efectiva en el proceso de aseguramiento de calidad del software. Preparar un informe que resuma los hallazgos y conclusiones del análisis.</p> <p>3 hrs. Aula 2 hrs. Independientes</p>	<p>Tipo de actividad: Aula (X) Plataforma() Laboratorio () Grupal (X) Individual () Equipo () Independientes (X)</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Pressman, R. S. (2010). "Ingeniería del Software, Un enfoque práctico" (7ma. ed). McGraw-Hill. • Documentación oficial de SonarQube y Checkstyle. <p>Criterios de evaluación de la actividad:</p> <p>Rúbrica de cuadro comparativo.</p>
<p>EC1 Fase II: Integración en el proceso de desarrollo de sistemas.</p>	
<p>Contenido: Integración de los conceptos y herramientas de aseguramiento de calidad del software en el proceso de desarrollo de sistemas.</p>	
<p>EC1 F2 Actividad de aprendizaje 3: Trabajo de investigación sobre aseguramiento de calidad del software.</p> <p>Diseñar, en equipos, un plan detallado para integrar los conceptos y herramientas de aseguramiento de calidad del software en el proceso de desarrollo de sistemas.</p>	<p>Tipo de actividad: Aula (X) Plataforma(X) Laboratorio () Grupal (X) Individual () Equipo () Independientes ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • ISO/IEC 29110: Ingeniería de software - Perfiles de ciclo de vida para microorganizaciones. • Pressman, R. S. (2010). "Ingeniería del Software, Un

<p>Para ello, es necesario:</p> <p>Definir los objetivos del plan:</p> <p>Establecer los objetivos específicos que se desean alcanzar con la implementación del aseguramiento de calidad del software.</p> <p>Especificar actividades concretas:</p> <p>Describir las actividades específicas que se realizarán para asegurar la calidad del software, como revisiones de código, pruebas automatizadas, y auditorías de calidad.</p> <p>Asignar responsabilidades:</p> <p>Distribuir las tareas entre los miembros del equipo, especificando quién será responsable de cada actividad.</p> <p>Elaborar un cronograma de implementación:</p> <p>Crear un calendario detallado que incluya las fechas y plazos para la realización de cada actividad.</p> <p>Presentar el plan en clase:</p> <p>Preparar una presentación en la que se exponga el plan diseñado, y estar listos para discutir posibles mejoras y desafíos.</p> <p>Discutir mejoras y desafíos:</p> <p>Durante la presentación, recibir retroalimentación de los compañeros y discutir en equipo cómo mejorar el plan y superar los desafíos identificados.</p> <p>Entrega:</p> <p>Entregar en plataforma.</p> <p>Evaluación:</p> <p>Cumplir con la rúbrica de trabajo de investigación.</p> <p>3 hrs. Aula 2 hrs. Plataforma</p>	<p>enfoque práctico" (7ma. ed). McGraw-Hill.</p> <ul style="list-style-type: none"> • IEEE Computer Society. (Última edición). Estándares de IEEE para el aseguramiento de calidad del software. • Guías de buenas prácticas de aseguramiento de calidad del software de instituciones reconocidas en la industria, como el SEI (Software Engineering Institute) y el CMMI (Capability Maturity Model Integration). <p>Criterios de evaluación de la actividad:</p> <p>Rúbrica de trabajo de investigación.</p>
<p>EC1 F2 Actividad de aprendizaje 4: Simulación de revisión de código con herramientas de análisis estático</p> <p>Realizar una simulación de revisión de código utilizando herramientas de análisis estático como SonarQube o Checkstyle. Los estudiantes deben formar equipos y revisar el código proporcionado utilizando estas herramientas. Identificar y discutir</p>	<p>Tipo de actividad:</p> <p>Aula (X) Plataforma(X) Laboratorio () Grupal () Individual () Equipo () Independientes (X)</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Documentación oficial de SonarQube y Checkstyle. • Pressman, R. S. (2010). "Ingeniería del Software, Un

<p>los posibles problemas de calidad del software encontrados durante la revisión.</p> <p>6 hrs. Aula 2 hrs. Plataforma 2 hrs. Independientes</p>	<p>enfoque práctico" (7ma. ed). McGraw-Hill.</p> <p>Criterios de evaluación de la actividad:</p> <p>Rúbrica de trabajo en equipo.</p>
<p>EC1 F2 Actividad de aprendizaje 5: Evaluación</p> <p>Realizar la evaluación correspondiente al primer elemento de competencia.</p> <p>2 hrs. Aula</p>	<p>Tipo de actividad: Aula (X) Plataforma() Laboratorio () Grupal () Individual (X) Equipo () Independientes ()</p> <p>Recursos:</p> <ul style="list-style-type: none"> • Pressman, R. S. (2010). "Ingeniería del Software, Un enfoque práctico" (7ma. ed). McGraw-Hill. • IEEE Computer Society. (Última edición). Estándares de IEEE para el aseguramiento de calidad del software. • ISO/IEC 29110: Ingeniería de software - Perfiles de ciclo de vida para microorganizaciones. • Guías de buenas prácticas de aseguramiento de calidad del software de instituciones reconocidas en la industria, como el SEI (Software Engineering Institute) y el CMMI (Capability Maturity Model Integration). • Documentación oficial de SonarQube y Checkstyle. <p>Criterios de evaluación de la actividad:</p> <p>Evaluación examen EC1</p>
<p>Evaluación formativa:</p> <p>Presentación sobre conceptos esenciales del aseguramiento de calidad del software</p> <p>Análisis de herramientas de análisis estático</p> <p>Trabajo de investigación sobre aseguramiento de calidad del software.</p> <p>Simulación de revisión de código con herramientas de análisis estático.</p> <p>Evaluación.</p>	
<p>Fuentes de información</p>	
<p>Pressman, R. S. (2010). "Ingeniería del Software, Un enfoque práctico" (7ma. ed). McGraw-Hill.</p> <p>IEEE Computer Society. (Última edición). Estándares de IEEE para el aseguramiento de calidad del software.</p> <p>ISO/IEC 29110: Ingeniería de software - Perfiles de ciclo de vida para microorganizaciones.</p> <p>Guías de buenas prácticas de aseguramiento de calidad del software de instituciones reconocidas en la industria, como el SEI (Software Engineering Institute) y el CMMI (Capability Maturity Model Integration).</p>	

Documentación oficial de SonarQube y Checkstyle.

Elemento de competencia 2: Comprender la estructura y los objetivos de los estándares y normas relacionados con la calidad del software, para liderar su implementación de manera responsable y efectiva en el proceso de desarrollo de sistemas, utilizando herramientas de gestión de pruebas.

Competencias blandas a promover: Liderazgo, resolución de problemas, flexibilidad.

EC2 Fase I: Exploración de estándares y herramientas de pruebas

Contenido: Estándares de calidad de software, fundamentos de herramientas de gestión de pruebas, proceso de prueba y comparación de herramientas.

EC2 F1 Actividad de aprendizaje 6: Análisis de caso: uso de herramientas de pruebas

Realizar en equipo pruebas comparativas entre diferentes herramientas de gestión de pruebas, como Selenium y JUnit. Cada equipo seleccionará un líder para dirigir el proceso de prueba y resolver los problemas que puedan surgir durante la ejecución de las pruebas. Después de la prueba, los equipos se reunirán para discutir los resultados, identificar posibles mejoras y ajustar su enfoque según sea necesario. Cada equipo mencionará en clase los resultados obtenidos.

1 hr. Aula
1 hr. Independiente

Tipo de actividad:

Aula (X) Plataforma() Laboratorio ()
Grupal (X) Individual () Equipo ()
Independientes (X)

Recursos:

Documentación oficial de [Selenium](#) y [JUnit](#).
Casos de prueba proporcionados por el instructor.

Criterios de evaluación de la actividad:

[Rúbrica de análisis de casos.](#)

EC2 F1 Actividad de aprendizaje 7: Implementación práctica de estándares y herramientas.

Implementar en equipo estándares y normas de calidad del software en un proyecto de desarrollo de sistemas real. Cada equipo seleccionará un líder para dirigir la implementación y resolver los problemas que puedan surgir durante el proceso. Se animará a los equipos a ser flexibles y adaptar los estándares según sea necesario para satisfacer las necesidades del proyecto. Al finalizar la implementación, los equipos evaluarán los resultados y compartirán lecciones aprendidas con el resto de la clase.

2 hrs. Aula
1 hr. Independiente

Tipo de actividad:

Aula (X) Plataforma() Laboratorio ()
Grupal (X) Individual () Equipo ()
Independientes (X)

Recursos:

Documentación oficial de los estándares seleccionados.
Tutoriales de [Selenium](#) y [JUnit](#).
Acceso al código fuente del proyecto seleccionado.

Criterios de evaluación de la actividad:

[Rúbrica de trabajo en equipo.](#)

EC2 Fase II: Evaluación y retroalimentación

Contenido: Implementación de estándares y normas de calidad del software, adaptación de estándares, pruebas prácticas, evaluación de resultados.

EC2 F2 Actividad de aprendizaje 8: Presentación sobre la evaluación de la implementación.

Presentar una evaluación de la implementación de

Tipo de actividad:

Aula (X) Plataforma() Laboratorio ()
Grupal (X) Individual () Equipo (X)
Independientes (X)

<p>los estándares y normas en su proyecto, identificando los logros alcanzados y los desafíos encontrados durante el proceso. Luego, los líderes de equipo presentarán sus resultados en clase y liderarán una discusión sobre las lecciones aprendidas y las áreas de mejora. Se fomentará la flexibilidad al considerar diferentes enfoques y soluciones para futuras implementaciones.</p> <p>4 hrs. Aula 1 hr. Independiente</p>	<p>Recursos:</p> <p>Documentación oficial de ISO 25000 .</p> <p>Webinar SonarQube .</p> <p>Artículo académico: "Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles " .</p> <p>Criterios de evaluación de la actividad: Rúbrica de presentación oral .</p>
<p>EC2 F2 Actividad de aprendizaje 9: Análisis de casos: implementación de estándares y normas de calidad de software.</p> <p>Analizar críticamente por equipos su implementación de estándares y normas de calidad del software. Identificarán áreas de mejora y propondrán acciones concretas para abordarlas en futuros proyectos. Cada equipo preparará un breve informe que incluya sus hallazgos y recomendaciones para compartir en una sesión grupal de retroalimentación.</p> <p>1 hr. Aula 5 hrs. Independientes</p>	<p>Tipo de actividad: Aula (X) Plataforma() Laboratorio () Grupal () Individual () Equipo () Independientes (X)</p> <p>Recursos:</p> <p>Documentación de la implementación (Actividad de aprendizaje 8).</p> <p>Pautas de retroalimentación proporcionadas por el instructor.</p> <p>Criterios de evaluación de la actividad: Rúbrica de análisis de casos .</p>
<p>EC2 F2 Actividad de aprendizaje 10: Evaluación: plan de acción para la implementación de estándares y normas de calidad del software.</p> <p>Diseñar un plan de acción para aplicar lo aprendido en proyectos futuros. Para ello, deberán identificar las áreas donde pueden aplicar el liderazgo, la resolución de problemas y la flexibilidad para mejorar la calidad del software en su trabajo futuro. Cada estudiante preparará un plan individual y lo presentará en una sesión de clase para compartir ideas y recibir retroalimentación; este trabajo será considerado como la evaluación del elemento.</p> <p>3 hrs. Aula 2 hrs. Plataforma 2 hrs. Independientes</p>	<p>Tipo de actividad: Aula (X) Plataforma(X) Laboratorio () Grupal (X) Individual (X) Equipo () Independientes (X)</p> <p>Recursos:</p> <p>Artículo académico: Propuesta para la mejora de la calidad del software en una consultora de tecnologías de la información aplicando las áreas prácticas del estándar CMMI DEV. 2.0 Nivel 2 .</p> <p>Criterios de evaluación de la actividad: Rúbrica de redacción de textos .</p>
<p>Evaluación formativa:</p> <p>Análisis de caso: uso de herramientas de pruebas</p>	

Implementación práctica de estándares y herramientas

Presentación sobre la evaluación de la implementación.

Análisis de casos: implementación de estándares y normas de calidad de software.

Evaluación: plan de acción para la implementación de estándares y normas de calidad del software.

Fuentes de información

Documentación oficial de Selenium y JUnit:

Selenium. (2023, June 27). Selenium. Selenium. <https://www.selenium.dev/>

JUnit 5. (2019). Junit.org. <https://junit.org/junit5/>

Casos de prueba proporcionados por el instructor.

Tutoriales de Selenium y JUnit:

Geek QA. (2021). Automatización de pruebas con Selenium IDE | Curso paso a paso con ejemplo [YouTube Video]. In YouTube. <https://www.youtube.com/watch?v=KskSQI5ZPVM&abchannel=GeekQA>

makigas. (2016). ¿Qué es JUnit 4 y cómo lo configuro en mi IDE? [YouTube Video]. In YouTube. <https://www.youtube.com/watch?v=EokoVm3rtNQ&listPLTd5ehIj0goML37B7s9I9iN2zhJCfxJBC&abchannel=makigas>

Documentación oficial de ISO 25000:

NORMAS ISO 25000. (2022). Iso25000.com. <https://iso25000.com/index.php/normas-iso-25000>

Webinar SonarQube:

Argentesting. (2018). Análisis estático de código con SonarQube [YouTube Video]. In YouTube. <https://www.youtube.com/watch?v=VUzwlt2dnD4&abchannel=Argentesting>

Artículo académico: "Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles":

Vista de Herramientas y buenas prácticas para el aseguramiento de calidad de software con metodologías ágiles. (2024). Revista de Investigación, Desarrollo e Innovación. https://revistas.upc.edu.co/index.php/investigacion_duitama/article/view/3277/3487

Artículo académico: "Propuesta para la mejora de la calidad del software en una consultora de tecnologías de la información aplicando las áreas prácticas del estándar CMMI DEV. 2.0 Nivel 2":

Montes, G., Isaac, G., Saldaña, M., David, A., Toyama, S., & Koyuzumi, F. (s.f.). Propuesta para la mejora de la calidad del software en una consultora de tecnologías de la información aplicando las áreas prácticas del estándar CMMI DEV 2.0 Nivel 2 Item Type info/semantics/masterThesis. https://repositorioacademico.upc.edu.pe/bitstream/handle/10757/672181/Ganvini_MG.pdf?sequence=1&isAllowd=y

Artículo académico en inglés sobre aseguramiento de calidad del software:

Laporte, C. Y., & April, A. (2018). *Software quality assurance*. John Wiley & Sons. <https://books.google.es/books?hl=es&lr&idpKVFDwAAQBAJ&oifnd&pgPA3&dqquality+assurance+software&otsGe5ACbyba2&sigslw9s8ZgQVOnXuK-1Q->

[KE9054#v=onepage&qquality%20assurance%20software&ffalse](#)

Elemento de competencia 3: Aplicar técnicas de calidad del software que permitan medir su eficacia mediante métricas adecuadas, para desarrollar programas computacionales fiables y eficientes que cumplan con los requerimientos del usuario dentro de las organizaciones, utilizando herramientas de análisis dinámico.

Competencias blandas a promover: Pensamiento crítico, gestión del tiempo, resiliencia.

EC3 Fase I: Introducción a técnicas de calidad del software y herramientas de análisis dinámico

Contenido: Características y funcionalidades de herramientas de análisis dinámico. Selección y evaluación de herramientas de análisis dinámico.

EC3 F1 Actividad de aprendizaje 11: Cuadro sinóptico: Exploración de herramientas de análisis dinámico.

Realizar un cuadro sinóptico, en base a una investigación y exploración de las herramientas de análisis dinámico; el cuadro debe mostrar las características clave de cada herramienta, sus capacidades y limitaciones, y cómo se pueden aplicar para mejorar la calidad del software. La información obtenida será comentada en clase de manera grupal, determinando cuáles son las mejores herramientas para realizar este tipo de revisiones.

1 hr. Aula
2 hrs. Plataforma

Tipo de actividad:

Aula (X) Plataforma(X) Laboratorio ()
Grupal (X) Individual () Equipo ()
Independientes ()

Recursos:

- Sitio oficial de [Tenable vulnerability Management](#).
- Sitios oficiales de herramientas de análisis de código dinámico.

Criterios de evaluación de la actividad:

[Rúbrica de cuadro sinóptico.](#)

EC3 F1 Actividad de aprendizaje 12: Exposición de casos de estudio de análisis dinámico

Investigar casos de estudio donde se haya utilizado alguna herramienta de análisis dinámico en proyectos de software reales. Preparar una presentación donde se explique el contexto del proyecto, las herramientas utilizadas, los resultados obtenidos y las lecciones aprendidas. Presentar los casos de estudio en clase y generar una discusión grupal sobre las diferentes aplicaciones y resultados observados.

3 hrs. Aula
2 hrs. Plataforma

Tipo de actividad:

Aula (X) Plataforma(X) Laboratorio ()
Grupal () Individual () Equipo ()
Independientes ()

Recursos:

Artículos académicos y técnicos disponibles en internet.

Criterios de evaluación de la actividad:

[Rúbrica de exposición.](#)

EC3 Fase II: Aplicación de técnicas de calidad del software en proyectos reales.

Contenido: Implementación de técnicas de calidad de software, uso de herramientas de análisis dinámico.

EC3 F2 Actividad de aprendizaje 13: Implementación de técnicas de calidad del software.

Aplicar, en equipos, técnicas de calidad del software, como pruebas de unidad, integración y sistema, en un proyecto de desarrollo de software real. Utilizarán herramientas de análisis dinámico

Tipo de actividad:

Aula (X) Plataforma() Laboratorio ()
Grupal (X) Individual () Equipo (X)
Independientes ()

Recursos:

Herramientas de análisis dinámico identificadas en la

<p>seleccionadas previamente para medir la eficacia de las técnicas aplicadas. Cada equipo presentará sus resultados ante el grupo.</p> <p>6 hrs. Aula</p>	<p>actividad anterior.</p> <p>Criterios de evaluación de la actividad: Rúbrica de trabajo en equipo .</p>
<p>EC3 F2 Actividad de aprendizaje 14: Mesa redonda: Taller de pruebas automatizadas</p> <p>Participar en un taller donde se enseñarán técnicas de automatización de pruebas utilizando herramientas específicas. Seleccionar una parte del proyecto de software real y aplicar pruebas automatizadas, documentando el proceso y los resultados. Realizar una sesión de retroalimentación para discutir las experiencias y los hallazgos.</p> <p>3 hrs. Aula</p>	<p>Tipo de actividad: Aula (X) Plataforma() Laboratorio () Grupal () Individual () Equipo () Independientes ()</p> <p>Recursos: Herramientas de pruebas automatizadas disponibles en internet.</p> <p>Criterios de evaluación de la actividad: Rúbrica de mesa redonda .</p>
<p>EC3 Fase III: Evaluación y mejora continua de la calidad del software</p> <p>Contenido: Evaluación de la eficacia de técnicas de calidad de software, mejora continua en el proceso de desarrollo de software.</p>	
<p>EC3 F3 Actividad de aprendizaje 15: Evaluación de la eficacia de las técnicas de calidad del software.</p> <p>Realizar una evaluación de la eficacia de las técnicas de calidad del software implementadas en el proyecto real elaborado anteriormente. Utilizando métricas adecuadas proporcionadas por las herramientas de análisis dinámico para medir el rendimiento y la fiabilidad del software desarrollado. La información obtenida deberá ser representada en un medio de representación visual de su elección (infografía, mapa conceptual, mapa mental, entre otros).</p> <p>4 hrs. Plataforma 2 hrs. Independientes</p>	<p>Tipo de actividad: Aula () Plataforma(X) Laboratorio () Grupal () Individual () Equipo () Independientes (X)</p> <p>Recursos:</p> <ul style="list-style-type: none"> Herramientas de análisis dinámico implementadas previamente. Canva. Cmaps Tools . <p>Criterios de evaluación de la actividad: Según corresponda:</p> <ul style="list-style-type: none"> Rúbrica de mapa mental . Rúbrica de infografía . Entre otros (según tipo de representación visual realizada).
<p>EC3 F3 Actividad de aprendizaje 16: Debate: retroalimentación y mejora continua.</p> <p>Revisar los resultados de la evaluación y discutir posibles mejoras en las técnicas de calidad del software y su aplicación en futuros proyectos. Mediante un debate grupal, identificarán áreas de mejora y propondrán acciones concretas para abordarlas.</p>	<p>Tipo de actividad: Aula (X) Plataforma() Laboratorio () Grupal (X) Individual () Equipo () Independientes (X)</p> <p>Recursos: Rúbrica de debate grupal .</p>

<p>1 hr. Aula 2 hrs. Independientes</p>	<p>Criterios de evaluación de la actividad: Rúbrica de debate.</p>	
<p>EC3 F3 Actividad de aprendizaje 17: Proyecto integrador</p> <p>Asegurar la calidad de un proyecto de software real, aplicando en todo momento los conceptos y herramientas vistos durante el curso. El docente proporcionará los detalles específicos sobre la elaboración del proyecto final.</p> <p>4 hrs. Aula 1 hr. Plataforma 10 hrs. Independientes</p>	<p>Tipo de actividad: Aula (X) Plataforma(X) Laboratorio () Grupal () Individual () Equipo () Independientes (X)</p> <p>Recursos: Casos prácticos vistos durante el curso.</p> <p>Criterios de evaluación de la actividad: Rúbrica de proyecto integrador.</p>	
<p>Evaluación formativa:</p> <p>Cuadro sinóptico: Exploración de herramientas de análisis dinámico.</p> <p>Exposición de casos de estudio de análisis dinámico</p> <p>Mesa redonda: Taller de pruebas automatizadas</p> <p>Implementación de técnicas de calidad del software.</p> <p>Evaluación de la eficacia de las técnicas de calidad del software.</p> <p>Debate: retroalimentación y mejora continua</p> <p>Proyecto integrador.</p>		
<p>Fuentes de información</p>		
<p><i>Tenable Vulnerability Management.</i> (2023). Tenable®. https://www.tenable.com/products/vulnerability-management</p> <p>Software recomendado:</p> <p>Canva.</p> <p>Cmaps Tools .</p> <p>Casos prácticos vistos durante el curso.</p>		
<p>Políticas</p> <p>Los teléfonos celulares deberán permanecer sin sonido en la clase</p>	<p>Metodología</p> <p>El curso se llevará a cabo mediante actividades en el aula durante las</p>	<p>Evaluación</p> <p>Artículo 27. La evaluación es el proceso que permite valorar el</p>

<p>y apagados durante los exámenes. Mostrar respeto dentro del aula para todos sus compañeros. No usar gafas para sol en el aula. Las inasistencias serán justificadas, mediante comprobante oficial o a criterio del profesor, siempre y cuando el estudiante se haya comunicado oportunamente y se establezcan las estrategias de recuperación de los temas tratados y de las actividades asignadas o desarrollada. Se permitirá una tolerancia de 10 minutos para entrar a clases. Será responsabilidad del estudiante Revisar la plataforma educativa con regularidad. Entregar las actividades en la plataforma educativa en la hora y fecha establecidas. Cumplir con todos los criterios de evaluación de la actividad.</p>	<p>clases presenciales y de forma virtual en la Plataforma Educativa Institucional. El docente les proporcionará un calendario de elaboración de actividades, que contemple las fechas específicas de entrega de cada actividad. Las actividades de aprendizaje serán entregadas por medio de la plataforma institucional, siguiendo las especificaciones del facilitador. Cualquier duda que se tenga acerca de los contenidos o asignaciones, es importante que se expongan al profesor, para no limitar su participación y aprendizaje. Durante el desarrollo del curso el alumno deberá participar activamente en las actividades que se enmarquen en ambos tipos de sesiones, con el fin de lograr la competencia establecida en cada elemento.</p>	<p>desarrollo de las competencias establecidas en las secuencias didácticas del plan de estudio del programa educativo correspondiente. Su metodología es integral y considera diversos tipos de evidencias de conocimiento, desempeño y producto por parte del alumno. Artículo 28. Las modalidades de evaluación en la Universidad son: I. Diagnóstica permanente, entendiéndola esta como la evaluación continua del estudiante durante la realización de una o varias actividades; II. Formativa, siendo esta, la evaluación al alumno durante el desarrollo de cada elemento de competencia; y III. Sumativa es la evaluación general de todas y cada una de las actividades y evidencias de las secuencias didácticas. Sólo los resultados de la evaluación sumativa tienen efectos de acreditación y serán reportados al departamento de registro y control escolar. Artículo 29. La evaluación sumativa será realizada tomando en consideración de manera conjunta y razonada, las evidencias del desarrollo de las competencias y los aspectos relacionados con las actitudes y valores logrados por el alumno. Para tener derecho a la evaluación sumativa de las asignaturas, el alumno deberá: I. Cumplir con la evidencia de las actividades establecidas en las secuencias didácticas; II. Asistir como mínimo al 70% de las sesiones de clase impartidas. Artículo 30. Los resultados de la evaluación expresarán el grado de dominio de las competencias, por lo que la escala de evaluación contemplará los niveles de: Competente sobresaliente; 1. Competente avanzado; 2. Competente intermedio; 3. Competente básico; y 4. No aprobado</p>
--	--	---