



Programa de Estudios de la Unidad de Aprendizaje: PRUEBAS DE SOFTWARE																													
Clave:	5FP-FM1172					Créditos:	3.37					Programa Académico:	TÉCNICO EN PROGRAMACIÓN																
Ramas de Conocimiento										Unidades Académicas donde se Imparte:																			
Ingeniería y Ciencias Físico Matemáticas	<input checked="" type="checkbox"/>		Ciencias Sociales Administrativas			Ciencias Médico Biológicas			TODAS LAS U.A.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	CET1
Área de Formación Curricular										Tiempos Asignados:																			
Institucional			Científica, Humanística y Tecnológica Básica			Profesional	<input checked="" type="checkbox"/>		Global: <u>54</u> Hrs/18 semanas/Semestre																				
Tipo de Espacio										Aula: <u>1</u> Hrs/Semana Total: <u>18</u> Hrs/Semestre																			
Aula	<input checked="" type="checkbox"/>		Taller			Laboratorio	<input checked="" type="checkbox"/>		Otros ambientes de Aprendizaje			Taller: <u>-</u> Hrs/Semana Total: <u>-</u> Hrs/Semestre																	
Modalidad										Laboratorio: <u>2</u> Hrs/Semana Total: <u>36</u> Hrs/Semestre																			
Escolarizada	<input checked="" type="checkbox"/>		No Escolarizada			Mixta			Otros ambientes de aprendizaje: <u>-</u> Hrs/Semana Total: <u>-</u> Hrs/Semestre																				
Vigencia a Partir:	AGOSTO 2024																												
Proceso de Diseño y Autorización:										Organización																			
					Día	Mes	Año	Por Unidad de Aprendizaje:	<input checked="" type="checkbox"/>					Por Área:						Por Módulo:									
Elaborado por: REP. ACAD. NMS										Firma y Sello de Autorización:																			
					Fecha de Elaboración:	09	11	2023																					
					Fecha de Revisión:	14	05	2024																					
					Fecha de Aprobación:	21	05	2024																					
					Fecha de Autorización:	23	05	2024																					
Autorizado por: CPA-CGC										M. EN E.N.A. MARÍA ISABEL ROJAS RUÍZ																			
										Directora de Educación Media Superior																			

## Programa Académico: Técnico en Programación

## Unidad de Aprendizaje: Pruebas de Software

### FUNDAMENTACIÓN

La Unidad de Aprendizaje **Pruebas de Software** pertenece al área de formación profesional del Bachillerato Tecnológico Bivalente del Nivel Medio Superior del Instituto Politécnico Nacional, se ubica en el quinto nivel del Plan de Estudios del Programa Académico Técnico en Programación y se imparte en la modalidad escolarizada de manera optativa en la rama del conocimiento de la Ingeniería y Ciencias Físico Matemáticas, desarrollando habilidades de acuerdo con las innovaciones y avances tecnológicos.

El propósito principal de la unidad de aprendizaje es que el estudiante realice pruebas a las aplicaciones de software a través del diseño de casos de prueba, técnicas, herramientas para la evaluación de la calidad del software de forma organizada, planificada, ética y responsable. Tiene como finalidad dar a conocer la importancia de la planificación y aplicación de las pruebas del software, el objetivo que busca cada prueba: unitaria, integración, regresión, sistema y aceptación, así como sus elementos que la conforman, las técnicas que se usan, el ciclo de vida de pruebas, el plan de pruebas, así como la evaluación de las cualidades del software funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

Desde un enfoque didáctico, esta unidad de aprendizaje propone el desarrollo integral del estudiante potenciando las habilidades cognitivas y socioemocionales generando experiencias de aprendizaje y de solución de problemas relacionados con las inconsistencias en el software que se generan a partir de errores lógicos, errores de sintaxis y fallas debido a configuraciones incorrectas o incompatibilidad por versiones de elementos que integran el software.

Así mismo, debe saber aplicar metodologías activas como: Trabajo colaborativo, Método del caso, Aprendizaje Basado en el Pensamiento, Métodos ágiles, Design Thinking, Aprendizaje Basado en Problemas, favoreciendo el desarrollo del talento a través del uso de las nuevas tecnologías de la información y comunicación que permitan a su vez al estudiante desarrollarse a nivel personal y profesional de forma continua a lo largo de la vida.

Por lo anterior, el docente que imparta esta unidad de aprendizaje Pruebas de Software debe dominar las habilidades y los conocimientos a desarrollar en el estudiante, de manera que pueda contribuir a su formación integral y desarrollar en ellos las competencias esenciales, seleccionar los métodos de enseñanza más adecuados, generar experiencias enriquecedoras y utilizar diversos ambientes de aprendizaje, atendiendo al desarrollo de aprendizajes conceptuales, procedimentales y actitudinales que permitan generar una educación inclusiva, flexible, sustentable y con perspectiva de género.

El rol del estudiante debe ser activo, participativo, inclusivo; se enfoca a la resolución de problemas reales, es autogestivo, se autoevalúa, participa también de la coevaluación, y aprende por iniciativa y tomando en cuenta sus estilos de aprendizaje. Innova, es creativo y trabaja en colaboración.

El proceso de enseñanza aprendizaje está centrado en el estudiante, por lo que la participación de este debe ser activa y comprometida con las actividades individuales dentro y fuera del aula además de actuar con responsabilidad social y ambiental, con respeto e inclusión con sus compañeros, todo en el marco de una formación integral. El estudiante debe adaptarse a nuevos ambientes de aprendizaje que le permitan desarrollarse de forma integral con su entorno social y productivo.

La evaluación de los aprendizajes comprenderá tres momentos: evaluación diagnóstica, evaluación formativa y evaluación sumativa. La evaluación diagnóstica permitirá valorar el nivel de conocimientos y habilidades que posea el estudiante antes de comenzar a revisar los temas de la unidad de aprendizaje, a fin de hacer ajustes a la programación y establecer conexiones significativas con la propuesta de aprendizaje. La evaluación formativa se implementará a lo largo del semestre para conocer los niveles de avance en el desarrollo de las competencias y se complementará con la autoevaluación y la coevaluación de los mismos estudiantes, enfatizando la retroalimentación oportuna. Por último, la evaluación sumativa se utilizará para valorar el grado en que el estudiante adquirió los conocimientos y desarrolló las habilidades esperadas, así como para establecer una



## Programa Académico: Técnico en Programación

## Unidad de Aprendizaje: Pruebas de Software

calificación numérica del curso. En el tercer momento de la evaluación y con fines de acreditación, también se diseñarán diferentes estrategias para englobar los conocimientos adquiridos necesarios para la comprensión y adquisición de nuevos conocimientos y habilidades.

Las evidencias con las que se evaluará formativa y sumativamente a los estudiantes; mostrarán su saber hacer de manera reflexiva, utilizando el conocimiento que va adquiriendo durante el proceso didáctico para luego transferir ese aprendizaje a situaciones similares y diferentes, en contextos escolares, personales y sociales, es por esto que esta unidad de aprendizaje se apoya en la unidad de aprendizaje Laboratorio de Desarrollo de Software III donde se integran proyectos, reportes finales, prácticas y presentaciones, entre otras situaciones observables con base en criterios específicos previamente conocidos por los estudiantes.

Los instrumentos de evaluación le permitirán al docente guiar y apoyar a los estudiantes a través de los indicadores que sirven de referencia a los estudiantes para lograr los niveles de desempeño esperados, por medio de la retroalimentación y reconocimiento de sus necesidades académicas, avalando, fortaleciendo sus logros y competencias adquiridas, incluyendo la heteroevaluación, la autoevaluación y la coevaluación en la clase.

Con base en la flexibilidad curricular y el reconocimiento de aprendizajes múltiples, será posible aplicar una evaluación para acreditar que el estudiante posee los conocimientos estipulados en la unidad de aprendizaje Pruebas de Software, previo a su inicio. De esta forma, el programa de estudio tiene una **naturaleza normativa**, puesto que establece los estándares para la certificación de conocimientos, habilidades prácticas del área de formación, habilidades socioemocionales, actitudes y valores con base a la normatividad del Instituto Politécnico Nacional.

Para el logro de los cometidos de la unidad de aprendizaje es importante contar con tres docentes en total que estén presentes tanto en la parte teórica como práctica, de los cuales, un docente es titular y dos docentes son auxiliares o adjuntos, que estarán reforzando las actividades dentro de talleres y laboratorios. La importancia de contar con dicha cantidad de docentes en la actividad de laboratorio es con el objetivo de reforzar el aprendizaje significativo, así como de atender que se cumplan con las normas de seguridad e higiene que aseguren integridad física del estudiante, el correcto empleo de las herramientas de medición en las áreas relacionadas con los laboratorios del Programa Académico de Técnico en Programación.





DESCRIPCIÓN DE LA UNIDAD DE APRENDIZAJE

Unidad de Aprendizaje: PRUEBAS DE SOFTWARE		
Propósito de la Unidad de Aprendizaje		
Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.		
Unidad 1: FUNDAMENTOS DE LAS PRUEBAS DEL SOFTWARE		
Unidad de competencia	Aprendizajes esperados	Contenidos de aprendizaje
1. Interpreta la importancia de la calidad en el desarrollo de productos de software mediante el uso de normas y categorías de pruebas para establecer indicadores de las mismas.	1. Reconoce las métricas en el desarrollo de productos de software reconociendo la importancia de la calidad a través del estudio de error, fallo y defecto.	<p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>Contextualización de la calidad en el software               <ul style="list-style-type: none"> <li>Definición de ingeniería de pruebas</li> <li>Calidad en el software</li> <li>Error</li> <li>Fallo</li> <li>Defecto</li> <li>Pruebas del software y sus características</li> <li>Métricas de revisión</li> <li>Revisiones informales</li> <li>Revisiones formales</li> <li>Aseguramiento de la calidad</li> <li>Normas de la calidad (ISO 9126)</li> </ul> </li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>Identifica las técnicas de revisión para el aseguramiento de calidad en el software</li> <li>Práctica 1. Calidad en el software</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>Respeto las posturas y los comentarios de sus compañeros</li> <li>Trabaja colaborativamente</li> <li>Participa de forma ética y responsable en las actividades durante la clase.</li> </ul>
	2. Identifica los tipos de pruebas mediante el qué prueba y cómo lo prueba, determinando los indicadores de las mismas.	





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

		<ul style="list-style-type: none"> <li>○ Pruebas de sistema</li> <li>○ Proceso de depuración</li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Identifica objetivos e indicadores en las pruebas del software. para justificar la implementación de las pruebas en el software.</li> <li>• Práctica 2. Tipos de pruebas</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Trabaja colaborativamente de forma responsable.</li> <li>• Muestra pensamiento crítico y analítico</li> <li>• Se comunica efectivamente</li> <li>• Respeto las posturas y los comentarios de sus compañeros</li> </ul>
<b>Unidad 2: PLANIFICACIÓN Y DISEÑO DE LAS PRUEBAS</b>		
<b>Unidad de competencia</b>	<b>Aprendizajes esperados</b>	<b>Contenidos de aprendizaje</b>
<p>2. Desarrolla un plan de pruebas a través de casos de prueba e indicadores para encontrar errores fallos y defectos del producto de software.</p>	<p>1. Planea las pruebas del software considerando los requerimientos de ambiente de pruebas y un plan de pruebas para establecer un propósito, alcance, un calendario de aplicación, y evaluación.</p>	<p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>• Planificación de las pruebas               <ul style="list-style-type: none"> <li>○ Plan de pruebas</li> <li>○ Propósito y alcance</li> <li>○ Calendarización de aplicación de pruebas</li> <li>○ Calendarización de evaluación</li> <li>○ Requerimientos de ambiente de pruebas</li> <li>○ Glosario de términos</li> <li>○ Validación</li> <li>○ Verificación</li> </ul> </li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Realiza un plan de pruebas orientado a su proyecto para establecer una calendarización de la aplicación de las pruebas.</li> <li>• Práctica 3. Plan de pruebas</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Trabaja colaborativamente de forma responsable.</li> <li>• Muestra pensamiento crítico y analítico</li> <li>• Resuelve problemas</li> </ul>
	<p>2. Construye los casos de prueba mediante formatos e indicadores específicos acorde al tipo de prueba para encontrar errores, fallos o defectos.</p>	<p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>• Elaboración de casos de prueba               <ul style="list-style-type: none"> <li>○ Propósito del plan</li> </ul> </li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

		<ul style="list-style-type: none"> <li>○ Prerrequisitos</li> <li>○ Indicadores</li> <li>○ Valores de entrada</li> <li>○ Pasos de ejecución</li> <li>○ Resultado esperado</li> <li>○ Resultado obtenido</li> <li>○ Estado de la prueba</li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Realiza casos de prueba para cada tipo de prueba, con la finalidad de encontrar errores, fallos y defectos.</li> <li>• Práctica 4. Casos de prueba</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Trabaja colaborativamente y de forma responsable</li> <li>• Muestra pensamiento crítico y analítico</li> <li>• Resuelve problemas</li> </ul>
<b>Unidad 3: EJECUCIÓN DE PRUEBAS Y EVALUACIÓN DE LA CALIDAD</b>		
<b>Unidad de competencia</b>	<b>Aprendizajes esperados</b>	<b>Contenidos de aprendizaje</b>
<p>3. Evalúa la calidad del software mediante resultados obtenidos de la aplicación de pruebas para lograr aplicaciones de software aptas para su implementación.</p>	<p>1. Pone a prueba aplicaciones de software mediante entornos de prueba, herramientas y técnicas para identificar errores, fallos y defectos.</p>	<p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>• Aplicación de pruebas <ul style="list-style-type: none"> <li>○ Herramientas y técnicas de pruebas</li> <li>○ Proceso de ejecución de las pruebas unitarias</li> <li>○ Proceso de ejecución de las pruebas de integración</li> <li>○ Proceso de ejecución de las pruebas funcionalidad</li> <li>○ Pruebas de Sistema</li> <li>○ Proceso de ejecución de las pruebas de rendimiento</li> </ul> </li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Aplica pruebas de software para identificar errores, fallos y defectos en el mismo con el objetivo de garantizar su calidad y correcto funcionamiento.</li> <li>• Práctica 5. Aplicación de pruebas</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Trabaja colaborativamente.</li> <li>• Muestra pensamiento reflexivo, crítico y analítico</li> <li>• Se comunica efectivamente</li> <li>• Muestra responsabilidad personal y social</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

	<p>2. Argumenta los resultados de pruebas obtenidos a través de métricas para lograr identificar las cualidades del software que le proporciona la calidad.</p>	<ul style="list-style-type: none"> <li>• Respeta las posturas y los comentarios de sus compañeros</li> </ul> <p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>• Métricas en las cualidades del software             <ul style="list-style-type: none"> <li>○ Funcionalidad</li> <li>○ Confiabilidad</li> <li>○ Usabilidad</li> <li>○ Eficiencia</li> <li>○ Mantenibilidad</li> <li>○ Portabilidad</li> </ul> </li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Distingue los indicadores de la calidad del software para establecer valores de entrada y esperados.</li> <li>• Analiza los resultados de las métricas de calidad para establecer un diagnóstico.</li> <li>• Práctica 6. Métricas de calidad en el software</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Trabaja colaborativamente.</li> <li>• Muestra pensamiento reflexivo, crítico y analítico</li> <li>• Se comunica efectivamente</li> <li>• Muestra responsabilidad personal y social</li> <li>• Respeto las posturas y los comentarios de sus compañeros</li> </ul>
--	---	---





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

**MATRIZ DE VINCULACIÓN**

	Unidad de Competencia 1		Unidad de Competencia 2		Unidad de Competencia 3	
	AE 1	AE 2	AE 1	AE 2	AE 1	AE 2
<b>COMPETENCIAS PARA EL SIGLO XXI HABILIDADES BLANDAS Y SOCIOEMOCIONALES</b>						
Trabajo colaborativo	X	X	X	X	X	X
Pensamiento reflexivo					X	X
Resolución de problemas.	X		X	X		
Pensamiento crítico y analítico		X	X	X	X	X
Comunicación efectiva		X			X	X
Creatividad						
Responsabilidad	X				X	X
Ética	X				X	X
Respeto	X	X	X	X	X	X





## Programa Académico: Técnico en Programación

## Unidad de Aprendizaje: Pruebas de Software

### PERFIL DOCENTE

El profesor que imparta la Unidad de Aprendizaje Pruebas del Software contará con las habilidades en el manejo de los saberes disciplinares y/o profesionales, así como su disposición, autoridad y tolerancia en el manejo de grupos de aprendizaje. Por lo tanto, debe poseer las habilidades que favorezcan el desarrollo del talento en el siglo XXI.

#### Habilidades docentes en el desarrollo del Talento

En el campo de su especialización:

- Habilidades y conocimientos profesionales que se requiere para la impartición de la Unidad de Aprendizaje Pruebas del Software.
- Adquirir habilidades digitales, desarrollarlas y actualizarlas.

En el campo pedagógico:

- Fomentar procesos de enseñanza que le permitan interpretar y resolver las necesidades de aprendizaje de los alumnos, tomando en cuenta sus capacidades, habilidades, vocación e intereses.
- Desarrollar procesos de enseñanza aprendizaje, utilizando métodos basados en administración de proyectos reales, aprovechando espacios educativos distintos a las aulas, para mejorar la calidad y pertinencia de la enseñanza.

En el campo de la investigación:

- Fortalecer el trabajo académico a partir del aprovechamiento de los resultados y productos de los proyectos de investigación

#### Perfil Profesional

- Titulado en Ing. en Sistemas Computacionales, Inteligencia Artificial, Ciencia de Datos, y/o Licenciado en Informática, Comunicaciones y Electrónica, Telemática, Computación o Maestría en Ciencias de la Computación, Sistemas Computacionales, Maestría en Gestión de la innovación en Tecnologías de la información y comunicación, maestría o especialización en docencia o afín con experiencia de dos años en el área docente.
- Experiencia comprobable de tres años en la iniciativa pública o privada aplicando los conocimientos de la unidad de aprendizaje

La unidad de aprendizaje Pruebas de software provee al estudiante de competencias profesionales que deben supervisarse y evaluarse individualmente, por la complejidad de la unidad de aprendizaje y el tamaño de los grupos, el docente titular requiere del apoyo de un docente auxiliar con las mismas competencias capaz de adecuar estrategias y recursos didácticos a objetivos específicos, para posteriormente evaluar y realimentar a los estudiantes al interior del laboratorio, así que cada docente podrá atender a varios estudiantes del mismo grupo permitiéndole resolver las dudas, orientar su aprendizaje y calidad.



**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

**ESTRUCTURA DIDÁCTICA**

<b>Unidad Didáctica 1:</b>	Fundamentos de las pruebas del software	<b>Nivel:</b>	<b>QUINTO</b>
<b>Propósito General:</b>	Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.		
<b>Unidad de Competencia No 1:</b>	Interpreta la importancia de la calidad en el desarrollo de productos de software mediante el uso de normas y categorías de pruebas para establecer indicadores de las mismas.		
<b>Aprendizaje Esperado No 1:</b>	Reconoce las métricas en el desarrollo de productos de software reconociendo la importancia de la calidad a través del estudio de error, fallo y defecto.	<b>Tiempo estimado para obtener el Aprendizaje Esperado:</b>	<b>9 horas</b>
Contenidos de Aprendizaje			
Conceptuales	Procedimentales	Actitudinales	
<ul style="list-style-type: none"> <li>Contextualización de la calidad en el software               <ul style="list-style-type: none"> <li>Definición de ingeniería de pruebas</li> <li>Calidad en el software</li> <li>Error</li> <li>Fallo</li> <li>Defecto</li> <li>Pruebas del software y sus características</li> <li>Métricas de revisión</li> <li>Revisiones informales</li> <li>Revisiones formales</li> <li>Aseguramiento de la calidad</li> <li>Normas de la calidad (ISO 9126)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Identifica las técnicas de revisión para el aseguramiento de calidad en el software</li> <li>Práctica 1. Calidad en el software</li> </ul>	<ul style="list-style-type: none"> <li>Respeto las posturas y los comentarios de sus compañeros</li> <li>Trabaja colaborativamente</li> <li>Participa de forma ética y responsable en las actividades durante la clase.</li> </ul>	
Estrategia Didáctica y Ambiente de Aprendizaje			
<p><b>Estrategia didáctica:</b> Trabajo colaborativo</p> <p><b>Apertura:</b></p> <ul style="list-style-type: none"> <li>El docente con apoyo de una presentación electrónica o similar expone y contextualiza la importancia de la calidad en el software, muestra la definición de ingeniería de pruebas, la diferencia entre error, fallo y defecto, aborda las pruebas de software y sus características, define las métricas de revisión, indica las diferencias entre las revisiones informales y formales y comenta la importancia de que tienen las normas en el aseguramiento de la calidad.</li> </ul> <p><b>Desarrollo:</b></p> <ul style="list-style-type: none"> <li>El docente seleccionará el tema de la “Importancia de las pruebas de software en la sociedad”.</li> <li>Conformará equipos de trabajo colaborativo de tres a cinco integrantes.</li> <li>El docente deberá solicitar la elaboración de un ensayo con base en material de consulta previamente curado.</li> <li>El tiempo de trabajo deberá considerar la extensión para abordar el aprendizaje esperado.</li> <li>Los estudiantes, en cada uno de su equipo de trabajo, de forma interna expondrá los puntos de vista del autor que haya seleccionado.</li> <li>Una vez que los estudiantes hayan abordado los conceptos fundamentales de pruebas, elaborarán un ensayo debidamente estructurado que les permita llegar a una conclusión.</li> </ul>			





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

- El docente en el laboratorio planteará una práctica que se relacione los contenidos conceptuales y el tema del proyecto que los estudiantes elaboran en la unidad de aprendizaje Laboratorio de Desarrollo de Software III.

**Cierre:**

- En plenaria y en equipo expondrán la conclusión de su ensayo, sobre el tema “La importancia de las pruebas de software en la sociedad”
- Los estudiantes, para reforzar los conocimientos previamente adquiridos deben realizar la Práctica 1. Calidad en software.

Ambiente de Aprendizaje: Aula y Laboratorio

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Computadora de escritorio</li> <li>• Software de desarrollo</li> <li>• Procesador de textos</li> </ul> <p><b>Recursos Didácticos:</b></p> <ul style="list-style-type: none"> <li>• Formato de reconocimiento de conceptos básicos de pruebas</li> </ul>	<p>Ensayo sobre el tema la importancia de las pruebas</p>	<p><b>Instrumento de Evaluación:</b></p> <ul style="list-style-type: none"> <li>• Lista de cotejo</li> </ul> <p><b>Criterios de Evaluación de forma:</b></p> <ul style="list-style-type: none"> <li>• Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>• Redacta sin errores de sintaxis ni de ortografía.</li> <li>• Cita las fuentes de consulta.</li> </ul> <p><b>Criterios de evaluación de forma fondo:</b></p> <ul style="list-style-type: none"> <li>• Explica los antecedentes de las pruebas del software.</li> <li>• Analiza mediante la comparación de al menos tres autores o fuentes de consulta.</li> <li>• Consulta fuentes de confianza.</li> <li>• Concluye con un resumen donde expone su postura y puntos de vista.</li> <li>• Reconoce la importancia de la aplicación de las pruebas del software</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

<b>Unidad Didáctica 1:</b>	<b>FUNDAMENTOS DE LAS PRUEBAS DEL SOFTWARE</b>	<b>Nivel:</b>	<b>QUINTO</b>
<b>Propósito General:</b>	Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.		
<b>Unidad de Competencia No 1:</b>	Interpreta la importancia de la calidad en el desarrollo de productos de software mediante el uso de normas y categorías de pruebas para establecer indicadores de las mismas.		
<b>Aprendizaje Esperado No 2:</b>	Identifica los tipos de pruebas mediante el qué prueba y cómo lo prueba, determinando los indicadores de las mismas.	<b>Tiempo estimado para obtener el Aprendizaje Esperado:</b>	<b>9 horas</b>

**Contenidos de Aprendizaje**

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> <li>Estrategias para las pruebas del software                             <ul style="list-style-type: none"> <li>Técnica de prueba de caja blanca, camino básico y caja negra</li> <li>Proceso de pruebas</li> <li>Pruebas de unidad</li> <li>Pruebas de integración</li> <li>Pruebas de validación</li> <li>Pruebas de sistema</li> <li>Proceso de depuración</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Identifica objetivos e indicadores en las pruebas del software. para justificar la implementación de las pruebas en el software.</li> <li>Práctica 2. Tipos de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>Trabaja colaborativamente de forma responsable.</li> <li>Muestra pensamiento crítico y analítico</li> <li>Se comunica efectivamente</li> <li>Respeto las posturas y los comentarios de sus compañeros</li> </ul>

**Estrategia Didáctica y Ambiente de Aprendizaje**

**Estrategia Didáctica:** Aprendizaje Basado en el Pensamiento

**Apertura:**

- El docente en la hora teórica ejemplifica las estrategias para realizar las pruebas donde se incluyen las técnicas de caja blanca, camino básico y caja negra. Expondrá los diferentes procesos para ejecutar las pruebas tales como pruebas de unidad, integración, validación, sistema y el proceso de depuración, además realizará demostraciones de las técnicas de caja blanca y caja negra, de tal forma que el estudiante podrá identificar en que tipo de pruebas debe emplearlas cada una de estas técnicas.

**Desarrollo:**

- Los estudiantes deberán reconocer los objetivos de cada tipo de prueba, comparar y contrastar las técnicas de pruebas de caja blanca y caja negra para tomar la decisión en donde pueden o no aplicarlas.
- Para reforzar los contenidos previamente vistos, en el laboratorio utilizando una metodología ágil, orientada por el profesor, los estudiantes desarrollarán una práctica donde tendrán la oportunidad de reproducir los procesos de aplicación de dichas técnicas.
- Los estudiantes realizan la Práctica 2. Tipos de pruebas

**Cierre:**

- Para finalizar, los estudiantes tendrán que resolver un cuestionario con el tema de los tipos de pruebas y su proceso, y en plenaria comentar las respuestas, el docente deberá fomentar el pensamiento crítico de los estudiantes orientado a la reflexión.

Ambiente de Aprendizaje: Aula y Laboratorio





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Computadora de escritorio</li> <li>• Software de desarrollo</li> <li>• Procesador de textos</li> </ul> <p><b>Recursos Didácticos:</b></p> <ul style="list-style-type: none"> <li>• Formato de elección de pruebas</li> </ul>	<p>Cuestionario de tipos de pruebas y su proceso.</p>	<p><b>Instrumento de Evaluación:</b></p> <ul style="list-style-type: none"> <li>• Lista de cotejo</li> </ul> <p><b>Criterios de evaluación de forma:</b></p> <ul style="list-style-type: none"> <li>• Participa cuando se le indica.</li> <li>• Respeta la postura de sus compañeros.</li> <li>• Contesta todas las preguntas.</li> <li>• Entrega en tiempo y forma lo solicitado</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Identifica de manera correcta los tipos de prueba de unidad, integración, validación, de sistema y depuración.</li> <li>• Reconoce la diferencia entre las técnicas de pruebas de caja blanca y caja negra.</li> <li>• Ubica el orden de la aplicación de las pruebas.</li> <li>• Identifica el proceso para ejecutar las pruebas tales como pruebas de unidad, integración, validación, sistema y depuración.</li> <li>• Identifica los elementos de las pruebas de unidad, integración, validación, sistema y depuración.</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

<b>Unidad Didáctica 2:</b>	Planificación y diseño de las pruebas	<b>Nivel:</b>	<b>Quinto</b>
<b>Propósito General:</b>	Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.		
<b>Unidad de Competencia No 2:</b>	Desarrolla un plan de pruebas a través de casos de prueba e indicadores para encontrar errores fallos y defectos del producto de software.		
<b>Aprendizaje Esperado No 1:</b>	Planea las pruebas del software considerando los requerimientos de ambiente de pruebas y un plan de pruebas para establecer un propósito, alcance, un calendario de aplicación, y evaluación.	<b>Tiempo estimado para obtener el Aprendizaje Esperado:</b>	<b>9 horas</b>
Contenidos de Aprendizaje			
Conceptuales	Procedimentales	Actitudinales	
<ul style="list-style-type: none"> <li>Planificación de las pruebas             <ul style="list-style-type: none"> <li>Plan de pruebas</li> <li>Propósito y alcance</li> <li>Calendarización de aplicación de pruebas</li> <li>Calendarización de evaluación</li> <li>Requerimientos de ambiente de pruebas</li> <li>Glosario de términos</li> <li>Validación</li> <li>Verificación</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Realiza un plan de pruebas orientado a su proyecto para establecer una calendarización de la aplicación de las pruebas.</li> <li>Práctica 3. Plan de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>Trabaja colaborativamente y de forma responsable</li> <li>Muestra pensamiento crítico y analítico</li> <li>Resuelve problemas</li> </ul>	
Estrategia Didáctica y Ambiente de Aprendizaje			
<p><b>Estrategia Didáctica:</b> Aprendizaje Basado en el Pensamiento</p> <p><b>Apertura:</b></p> <ul style="list-style-type: none"> <li>El docente explica el plan de pruebas y sus elementos, tales como propósito, alcance, calendarización de aplicación, calendarización de evaluación, requerimientos de ambiente pruebas y glosario de términos, además debe indicar la incidencia en los procesos de validación y verificación y mencionar la relevancia de un proyecto de software y las pruebas de unidad, integración, validación y de sistema.</li> </ul> <p><b>Desarrollo:</b></p> <ul style="list-style-type: none"> <li>El docente planteará una actividad que consiste en analizar la implementación de un plan de pruebas a un proyecto de software.</li> <li>Los estudiantes deben analizar e interiorizar la información o procedimiento para la elaboración del Plan de Pruebas descrito por el docente.</li> <li>Identificarán el propósito y alcance considerando las diferencias entre los tipos de pruebas según los objetivos de cada una.</li> <li>En plenaria los estudiantes deberán participar emitiendo su opinión en cuanto a la implementación de cada una de las pruebas.</li> <li>El estudiante elaborará un organizador gráfico que le permita identificar los pasos de los procesos de validación y verificación y la relación con el plan de pruebas.</li> </ul> <p><b>Cierre:</b></p> <ul style="list-style-type: none"> <li>El estudiante podrá explicar cada elemento de un plan de pruebas y su impacto en un proyecto de desarrollo de software al implementar las pruebas de unidad, integración, validación y sistema.</li> <li>Los estudiantes en el laboratorio realizarán la Práctica 3. Plan de pruebas</li> </ul>			





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Equipo de cómputo</li> <li>• Internet</li> <li>• Proyector</li> <li>• Repositorio digital.</li> <li>• Repositorios de contenidos y material de consulta</li> <li>• Software orientado a la unidad de competencia</li> </ul> <p><b>Recursos didácticos:</b></p> <ul style="list-style-type: none"> <li>• Infografías</li> <li>• Presentaciones electrónicas relacionadas al tema.</li> <li>• Páginas, sitios web relacionadas al tema.</li> <li>• Libros digitales o impresos relacionados al tema.</li> </ul>	<p>Presentación del organizador gráfico de los elementos del Plan de Pruebas</p>	<p><b>Instrumento de Evaluación:</b></p> <ul style="list-style-type: none"> <li>• Guía de observación</li> </ul> <p><b>Criterios de evaluación forma:</b></p> <ul style="list-style-type: none"> <li>• Es puntual, iniciando en tiempo su presentación.</li> <li>• Es fluido al hablar</li> <li>• Respeta el tiempo para exponer el plan de pruebas</li> <li>• Presenta a los integrantes del equipo.</li> <li>• Respeta los comentarios de sus compañeros.</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Explica el propósito del plan de pruebas.</li> <li>• Comenta el porqué del alcance</li> <li>• Menciona los términos utilizados, explicando cada uno de estos.</li> <li>• Explica la importancia del porque se deben aplicar cada una de las pruebas.</li> <li>• Indica los criterios para evaluar cada una de las pruebas.</li> <li>• Comenta los requerimientos del ambiente de pruebas.</li> <li>• Concluye indicando la importancia de la inclusión de un plan de pruebas en un proyecto de software.</li> <li>• Muestra pensamiento crítico y analítico</li> </ul>



**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

<b>Unidad Didáctica 2:</b>	Planificación y diseño de las pruebas	<b>Nivel:</b>	<b>Quinto</b>
<b>Propósito General:</b>	Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.		
<b>Unidad de Competencia No 2:</b>	Desarrolla un plan de pruebas a través de casos de prueba e indicadores para encontrar errores fallos y defectos del producto de software.		
<b>Aprendizaje Esperado No 2:</b>	Construye los casos de prueba mediante formatos e indicadores específicos acorde al tipo de prueba para encontrar errores, fallos o defectos.	<b>Tiempo estimado para obtener el Aprendizaje Esperado:</b>	<b>9 horas</b>

**Contenidos de Aprendizaje**

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> <li>Elaboración de casos de prueba               <ul style="list-style-type: none"> <li>Propósito del caso de prueba</li> <li>Prerrequisitos</li> <li>indicadores</li> <li>Valores de entrada</li> <li>Pasos de ejecución</li> <li>Resultado esperado</li> <li>Resultado obtenido</li> <li>Estado de la prueba</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Realiza casos de prueba para cada tipo de prueba, con la finalidad de encontrar errores, fallos y defectos.</li> <li>Práctica 4. Casos de prueba</li> </ul>	<ul style="list-style-type: none"> <li>Trabaja colaborativamente y de forma responsable</li> <li>Muestra pensamiento crítico y analítico</li> <li>Resuelve problemas</li> </ul>

**Estrategia Didáctica y Ambiente de Aprendizaje**

**Estrategia Didáctica:** Aprendizaje Basado en Problemas

**Apertura:**

El docente explica los elementos de los casos de prueba, las métricas, los indicadores, prerrequisitos, valores de entrada, pasos de ejecución, resultado esperado, resultado obtenido y estado de la prueba y ofrece ejemplos de su aplicación para los tipos de pruebas.

**Desarrollo:**

- El docente planteará un problema relacionado con la aplicación de casos de pruebas en un proyecto de software
- Los estudiantes de forma colaborativa analizarán el problema y las causas que lo provocan.
- Recabarán información para establecer propuestas de solución.
- Determinarán los posibles casos de prueba acorde a las pruebas requeridas.
- Los estudiantes con los conocimientos previamente adquiridos, identifican métricas, indicadores, prerrequisitos, valores de entrada, pasos de ejecución, resultado esperado, resultado obtenido y estado de la prueba.
- Establecerá la solución del problema usando los casos de prueba.
- Realiza una conclusión
- El docente guía y monitorea las acciones que han realizado los estudiantes.

**Cierre:**

- Los estudiantes en equipo y en plenaria presentan los casos de prueba elegidos para las pruebas de unidad, integración, validación y de sistema.
- Los estudiantes en equipo y en laboratorio trabajarán una práctica para realizar la elaboración de los casos de pruebas para los diferentes tipos de prueba: pruebas de unidad, integración, validación y de sistema orientados a una aplicación de software, preferiblemente orientada a su proyecto que desarrollan en la unidad de aprendizaje Laboratorio de Desarrollo de Software III.





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

- Los estudiantes elaboran la Práctica 4. Casos de prueba
- Ambiente de Aprendizaje: Laboratorio / Aula

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Equipo de cómputo</li> <li>• Internet</li> <li>• Proyector</li> <li>• Repositorio digital.</li> <li>• Repositorios de contenidos y material de consulta</li> <li>• Software orientado a la unidad de competencia</li> </ul> <p><b>Recursos didácticos:</b></p> <ul style="list-style-type: none"> <li>• Infografías</li> <li>• Presentaciones electrónicas relacionadas al tema.</li> <li>• Páginas, sitios web relacionadas al tema.</li> <li>• Libros digitales o impresos relacionados al tema.</li> </ul>	<p>Presentación de la solución con los casos de prueba</p>	<p><b>Instrumento de Evaluación:</b></p> <ul style="list-style-type: none"> <li>• Guía de observación</li> </ul> <p><b>Criterios de evaluación forma:</b></p> <ul style="list-style-type: none"> <li>• Es puntual, iniciando en tiempo su presentación.</li> <li>• Es fluido al hablar</li> <li>• Respeto el tiempo para exponer el plan de pruebas</li> <li>• Presenta a los integrantes del equipo.</li> <li>• Respeto los comentarios de sus compañeros.</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Explica la importancia de la inclusión de las pruebas de unidad, integración, validación, sistema y depuración.</li> <li>• Comenta los indicadores que se utilizarán para cada caso de prueba.</li> <li>• Menciona prerrequisitos a considerar en cada caso de prueba.</li> <li>• Indica los pasos a ejecutar para cada uno de los casos de las pruebas de unidad, integración, validación, sistema y depuración.</li> <li>• Muestra pensamiento crítico y analítico</li> </ul>



**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

<b>Unidad Didáctica 3:</b>	Ejecución de pruebas y evaluación de la calidad	<b>Nivel:</b>	<b>Quinto</b>
<b>Propósito General:</b>	Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.		
<b>Unidad de Competencia No 3:</b>	Evalúa la calidad del software mediante resultados obtenidos de la aplicación de pruebas para lograr aplicaciones de software aptas para su implementación.		
<b>Aprendizaje Esperado No 1:</b>	Pone a prueba aplicaciones de software mediante entornos de prueba, herramientas y técnicas para identificar errores, fallos y defectos.	<b>Tiempo estimado para obtener el Aprendizaje Esperado:</b>	<b>9 horas</b>

**Contenidos de Aprendizaje**

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> <li>• Aplicación de pruebas               <ul style="list-style-type: none"> <li>○ Herramientas y técnicas de pruebas</li> <li>○ Proceso de ejecución de las pruebas unitarias</li> <li>○ Proceso de ejecución de las pruebas de integración</li> <li>○ Proceso de ejecución de las pruebas funcionalidad</li> <li>○ Pruebas de Sistema</li> <li>○ Proceso de ejecución de las pruebas de rendimiento</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Aplica pruebas de software para identificar errores, fallos y defectos en el mismo con el objetivo de garantizar su calidad y correcto funcionamiento</li> <li>• Práctica 5. Aplicación de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>• Trabaja colaborativamente.</li> <li>• Muestra pensamiento reflexivo, crítico y analítico</li> <li>• Se comunica efectivamente</li> <li>• Muestra responsabilidad personal y social</li> <li>• Respeta las posturas y los comentarios de sus compañeros</li> </ul>

**Estrategia Didáctica y Ambiente de Aprendizaje**

**Estrategia Didáctica:** Aprendizaje Basado en el Pensamiento

**Apertura:**  
El docente en el aula de clase expone las herramientas y técnicas que se usan en el proceso de la ejecución de las pruebas unitarias, de integración, funcionalidad, pruebas de sistema y de rendimiento.

**Desarrollo:**  
El docente solicitará la elaboración de un cuadro comparativo con las características de las herramientas y técnicas para el proceso de ejecución de las pruebas unitarias, de integración, funcionalidad, pruebas de sistema y de rendimiento.  
El docente realizará una demostración de la instalación y aplicación del software para ejecutar las pruebas unitarias, de integración, funcionalidad, pruebas de sistema y de rendimiento.  
El estudiante conocerá las funciones de una librería o software para ejecutar pruebas unitarias y pruebas de sistema (sobrecarga y rendimiento).  
Reconocerá las diferencias entre las pruebas y el software apropiado para la ejecución de las diferentes pruebas.  
Elaborará un cuadro comparativo donde se aprecien que herramientas, técnicas son los adecuados para ejecutar las pruebas.

**Cierre**

- En equipos de trabajo socializará el cuadro comparativo para finalmente crear uno solo y presentarlo en una exposición.
- Los estudiantes realizan la Práctica 5. Aplicación de pruebas

Ambiente de Aprendizaje: Aula y Laboratorio





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Computadora de escritorio</li> <li>• Software de desarrollo</li> <li>• Librerías para pruebas unitarias</li> <li>• Software para hacer pruebas de sistema</li> <li>• Procesador de textos</li> </ul> <p><b>Recursos Didácticos:</b></p> <ul style="list-style-type: none"> <li>• Formato de Plan de pruebas</li> </ul>	<p>Cuadro comparativo de las herramientas, técnicas para la ejecución de las pruebas.</p>	<p><b>Instrumento de Evaluación:</b></p> <ul style="list-style-type: none"> <li>• Lista de cotejo</li> </ul> <p><b>Criterios de evaluación de forma:</b></p> <ul style="list-style-type: none"> <li>• Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>• Entrega en tiempo y forma.</li> <li>• Redacta sin errores de sintaxis ni de ortografía.</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Identifica la aplicación de caminos independientes.</li> <li>• Ubica la aplicación de aristas, nodos y grafo de flujo.</li> <li>• Asocia las herramientas necesarias para la aplicación de cada tipo de prueba.</li> <li>• Reconoce las librerías requeridas para las pruebas unitarias.</li> <li>• Reconoce el proceso de ejecución de las pruebas unitarias, de integración, funcionalidad, pruebas de sistema y de rendimiento.</li> <li>• Se aprecia la comparación entre las diferentes pruebas.</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

<b>Unidad Didáctica 3:</b>	Ejecución de pruebas y evaluación de la calidad	<b>Nivel:</b>	<b>Quinto</b>
<b>Propósito General:</b>	Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.		
<b>Unidad de Competencia No 3:</b>	Evalúa la calidad del software mediante resultados obtenidos de la aplicación de pruebas para lograr aplicaciones de software aptas para su implementación.		
<b>Aprendizaje Esperado No 2:</b>	Argumenta los resultados de pruebas obtenidos a través de métricas para lograr identificar las cualidades del software que le proporciona la calidad.	<b>Tiempo estimado para obtener el Aprendizaje Esperado:</b>	<b>9 horas</b>

**Contenidos de Aprendizaje**

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> <li>Métricas en las cualidades del software               <ul style="list-style-type: none"> <li>Funcionalidad</li> <li>Confiabilidad</li> <li>Usabilidad</li> <li>Eficiencia</li> <li>Mantenibilidad</li> <li>Portabilidad</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Distingue los indicadores de la calidad del software para establecer valores de entrada y esperados.</li> <li>Analiza los resultados de las métricas de calidad para establecer un diagnóstico.</li> <li>Práctica 6. Métricas de calidad en el software</li> </ul>	<ul style="list-style-type: none"> <li>Trabaja colaborativamente.</li> <li>Muestra pensamiento reflexivo, crítico y analítico</li> <li>Se comunica efectivamente</li> <li>Muestra responsabilidad personal y social</li> <li>Respeto las posturas y los comentarios de sus compañeros</li> </ul>

**Estrategia Didáctica y Ambiente de Aprendizaje**

**Estrategia Didáctica:** Aprendizaje Basado en el Pensamiento

**Apertura:**

- El docente con apoyo de un esquema gráfico muestra a los estudiantes las definiciones, características, elementos e indicadores y ejemplos de las métricas para medir la calidad del software tales como funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad. El docente mostrará ejemplos de aplicación para la evaluación de las cualidades del software.

**Desarrollo:**

- El docente planteara una situación para que el estudiante la pueda analizar
- El docente presentará una tabla con los nombres de cada una de las cualidades del software dejando un espacio para que el estudiante la complemente con los indicadores de la calidad que le sirvan para establecer valores de entrada y esperados.
- El estudiante comparará y contrastará la información dada por el docente para completar la tabla.

**Cierre:**

- En equipos de tres personas los estudiantes socializarán su trabajo, una vez hecho esto deberán socializar sus resultados y conclusiones en plenaria.
- Los estudiantes realizan la Práctica 6. Métricas de calidad en el software

Ambiente de Aprendizaje: Aula y laboratorio.





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Computadora de escritorio</li> <li>• Software de desarrollo</li> <li>• Software para hacer pruebas de sistema</li> <li>• Procesador de textos</li> </ul> <p><b>Recursos Didácticos:</b></p> <ul style="list-style-type: none"> <li>• Formato de métricas de calidad</li> </ul>	<p>Informe de los resultados de las métricas de calidad</p>	<p><b>Instrumento de Evaluación:</b> Lista de cotejo</p> <p><b>Criterios de evaluación de forma:</b></p> <ul style="list-style-type: none"> <li>• Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>• No tiene errores de ortografía</li> <li>• La redacción es clara y mantiene un tono profesional.</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Muestra los resultados obtenidos de las métricas de calidad para la evaluación de funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.</li> <li>• Analiza los resultados obtenidos evidenciando los errores fallos y defectos encontrados.</li> <li>• Interpreta los resultados obtenidos para cada métrica de calidad del software.</li> <li>• Establece un criterio a partir de los resultados obtenidos.</li> <li>• Proporciona recomendaciones para mejorar la calidad del software.</li> <li>• Evalúa la calidad del software partiendo de un análisis de resultados.</li> </ul>



**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

**PRÁCTICAS**

<b>Nombre de la Práctica:</b>	Calidad en el software	<b>N° de la Práctica:</b>	1	<b>Tiempo:</b>	6 horas
<b>Unidad de Competencia:</b>	Interpreta la importancia de la calidad en el desarrollo de productos de software mediante el uso de normas y categorías de pruebas para establecer indicadores de las mismas.				
<b>Aprendizajes Esperados Relacionados con la Práctica:</b>	Reconoce las métricas en el desarrollo de productos de software reconociendo la importancia de la calidad a través del estudio de error, fallo y defecto.				
Contenidos de Aprendizaje Relacionados con la Práctica					
Conceptuales	Procedimentales	Actitudinales			
<b>Conceptual:</b> <ul style="list-style-type: none"> <li>• Contextualización de la calidad en el software               <ul style="list-style-type: none"> <li>○ Definición de ingeniería de pruebas</li> <li>○ Calidad en el software</li> <li>○ Error</li> <li>○ Fallo</li> <li>○ Defecto</li> <li>○ Pruebas del software y sus características</li> <li>○ Métricas de revisión</li> <li>○ Revisiones informales</li> <li>○ Revisiones formales</li> <li>○ Aseguramiento de la calidad</li> <li>○ Normas de la calidad (ISO 9126)</li> </ul> </li> </ul>	Identifica las técnicas de revisión para el aseguramiento de calidad en el software	<ul style="list-style-type: none"> <li>• Respeta las posturas y los comentarios de sus compañeros</li> <li>• Trabaja colaborativamente</li> <li>• Participa de forma ética y responsable en las actividades durante la clase.</li> </ul>			
Estrategia Didáctica y Ambiente de Aprendizaje					
<b>Estrategia Didáctica:</b> Método del caso La práctica busca que los estudiantes fortalezcan los conceptos fundamentales de la ingeniería de pruebas, busquen información sobre la importancia de la aplicación de las pruebas que les permita analizar diversos casos para reflexionar sobre el aseguramiento de la calidad en el software.					
<ul style="list-style-type: none"> <li>• Fase de elaboración (Selección del caso): El docente deberá seleccionar los tipos de casos relacionados con el aseguramiento de la calidad en el software, así como las fuentes de consulta. Lis casos reales en los que el software haya fallado y provocado catástrofes o problemas al medio ambiente y en la sociedad.</li> <li>• Fase Preliminar: el docente presentará el contexto de la práctica, el tema abordar y las características de los casos a estudiar, preferentemente en quipos de trabajo.</li> <li>• Fase Eclósiva: Los estudiantes, participan de manera activa, realizando una búsqueda de información de los casos propuestos por el docente. Los estudiantes, internamente en su equipo de trabajo generan opiniones, impresiones y juicios.</li> <li>• Fase de Análisis: Los estudiantes realizan una búsqueda en común del sentido de los acontecimientos, se integran aspectos informativos y se integra una síntesis de los casos a estudiar.</li> <li>• Fase de conceptualización: Se formulan conceptos operativos o principios concretos de acción explicables en el caso que permitan aproximarse a la explicación o análisis o solución del caso.</li> <li>• Fase de evaluación: Se deben identificar los hechos, el problema, y la solución, así como la participación de los estudiantes.</li> </ul> En un documento electrónico elaborado con un procesador de textos el estudiante debe documentar un informe y con sus propias palabras dar respuesta a lo siguiente:					
<ul style="list-style-type: none"> <li>• El por qué de la importancia de las pruebas.</li> </ul>					



**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

- Mencionar y describir al menos tres de los principios de pruebas.
- Identificar los elementos de las revisiones formales e informales
- Describir y asociar y las técnicas de revisión para el aseguramiento de calidad en el software

Ambiente de Aprendizaje: Laboratorio

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Computadora personal</li> <li>• Conexión a internet</li> </ul> <p><b>Recursos Didácticos:</b></p> <ul style="list-style-type: none"> <li>• Bibliografía digital</li> </ul>	<p>Informe documental</p>	<p><b>Instrumento de Evaluación:</b></p> <ul style="list-style-type: none"> <li>• Guía de observación</li> </ul> <p><b>Criterios de evaluación forma:</b></p> <ul style="list-style-type: none"> <li>• Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>• Entrega en tiempo y forma</li> <li>• Redacta sin errores de sintaxis ni de ortografía.</li> <li>• Cita las fuentes de consulta</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Navega por sitios de índole académico y de confianza.</li> <li>• Investiga por lo menos tres casos.</li> <li>• Identifica fallas, catástrofes y problemas con la sociedad y medio ambiente causados por la mala calidad en el software.</li> <li>• Expone la importancia de la aplicación de las pruebas.</li> <li>• Menciona y describe al menos tres de los principios de pruebas.</li> <li>• Identifica los elementos de las revisiones formales e informales.</li> <li>• Describe las técnicas de revisión para el aseguramiento de calidad en el software</li> <li>• Socializa los resultados obtenidos con sus compañeros.</li> </ul>



**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

<b>Nombre de la Práctica:</b>	Tipos de pruebas	<b>N° de la Práctica:</b>	2	<b>Tiempo:</b>	6 horas
<b>Unidad de Competencia:</b>	Interpreta la importancia de la calidad en el desarrollo de productos de software mediante el uso de normas y categorías de pruebas para establecer indicadores de las mismas.				
<b>Aprendizajes Esperados Relacionados con la Práctica:</b>	Identifica los tipos de pruebas mediante el qué prueba y cómo lo prueba, determinando los indicadores de las mismas.				

**Contenidos de Aprendizaje Relacionados con la Práctica**

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> <li>• Estrategias para las pruebas del software               <ul style="list-style-type: none"> <li>○ Técnica de prueba de caja blanca, camino básico y caja negra</li> <li>○ Proceso de pruebas</li> <li>○ Pruebas de unidad</li> <li>○ Pruebas de integración</li> <li>○ Pruebas de validación</li> <li>○ Pruebas de sistema</li> <li>○ Proceso de depuración</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Identifica objetivos e indicadores en las pruebas del software. para justificar la implementación de las pruebas en el software.</li> <li>• Práctica 2. Tipos de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>• Trabaja colaborativamente de forma responsable.</li> <li>• Muestra pensamiento crítico y analítico</li> <li>• Se comunica efectivamente</li> <li>• Respeto las posturas y los comentarios de sus compañeros</li> </ul>

**Estrategia Didáctica y Ambiente de Aprendizaje**

**Estrategia Didáctica:** AGILE

El objetivo de la práctica es que el estudiante identifique los objetivos e indicadores en las pruebas del software para justificar la implementación de las pruebas como parte del desarrollo de un proyecto. El docente organizará equipos de trabajo con los estudiantes, usando una metodología ágil cada equipo debe ejecutar un sprint por cada tipo de prueba en el cual determinará el objetivo de la prueba, indicadores a utilizar y una justificación de la implementación para las pruebas de:

- Pruebas de unidad
- Pruebas de integración
- Pruebas de validación
- Pruebas de sistema
- Proceso de depuración

Los estudiantes, al finalizar todos los sprints deben realizar un informe con los resultados obtenidos en donde deben incluir los tipos de prueba y sus elementos.

Ambiente de Aprendizaje: Laboratorio

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p><b>Herramientas Tecnológicas:</b> Computadora personal Conexión a internet</p> <p><b>Recursos Didácticos:</b> Bibliografía digital</p>	<p>Informe de los sprints</p>	<p><b>Instrumento de Evaluación:</b></p> <ul style="list-style-type: none"> <li>• Guía de observación</li> </ul> <p><b>Criterios de evaluación forma:</b></p> <ul style="list-style-type: none"> <li>• Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>• Entrega en tiempo y forma</li> <li>• Redacta sin errores de sintaxis ni de ortografía</li> <li>• Integra las referencias bibliográficas</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Realiza adecuadamente un sprint por cada prueba.</li> <li>• Documenta los resultados obtenidos en cada uno de los sprints.</li> <li>• Identifica objetivos e indicadores para cada una de las pruebas</li> <li>• Justifica la implementación de cada una de las pruebas.</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

<b>Nombre de la Práctica:</b>	Plan de pruebas	<b>Nº de la Práctica:</b>	<b>3</b>	<b>Tiempo:</b>	<b>6 horas</b>
<b>Unidad de Competencia:</b>	Desarrolla un plan de pruebas a través de casos de prueba e indicadores para encontrar errores fallos y defectos del producto de software.				
<b>Aprendizajes Esperados</b>	Planea las pruebas del software considerando los requerimientos de ambiente de pruebas y un plan de pruebas para establecer un propósito, alcance, un calendario de aplicación, y evaluación.				
<b>Relacionados con la Práctica:</b>					

**Contenidos de Aprendizaje Relacionados con la Práctica**

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> <li>• Planificación de las pruebas                             <ul style="list-style-type: none"> <li>○ Plan de pruebas</li> <li>○ Propósito y alcance</li> <li>○ Calendarización de aplicación de pruebas</li> <li>○ Calendarización de evaluación</li> <li>○ Requerimientos de ambiente de pruebas</li> <li>○ Glosario de términos</li> <li>○ Validación</li> <li>○ Verificación</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Realiza un plan de pruebas orientado a su proyecto para establecer una calendarización de la aplicación de las pruebas.</li> <li>• Práctica 3. Plan de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>• Trabaja colaborativamente de forma responsable.</li> <li>• Muestra pensamiento crítico y analítico</li> <li>• Resuelve problemas</li> </ul>

**Estrategia Didáctica y Ambiente de Aprendizaje**

**Estrategia didáctica.** Design Thinking

La práctica tiene como objetivo que los estudiantes elaboren un plan de pruebas orientado al proyecto de software que realizan en la unidad de aprendizaje Laboratorio de Desarrollo de Software o alguno similar.

La elaboración del plan de pruebas deberá ser mediante el trabajo colaborativo y considerar los siguientes pasos:

1. El estudiante debe identificar el tipo de usuario al cual esta dirigida una aplicación de software para conocer sus características y necesidades.
2. Establecer un propósito y alcance para precisar los objetivos de las pruebas del software.
3. En equipo los estudiantes deberán discutir los alcances de cada una de las pruebas para definir una calendarización y establecer los requerimientos del ambiente de pruebas, así como crear un glosario de términos del contexto del proyecto.
4. El estudiante construye un prototipo del Plan de Pruebas que le permita visualizar el alcance de cada una y los tiempos de aplicación.
5. Los estudiantes presentan en plenaria el Plan de Pruebas para validar, depurar y mejorar lo ya planteado.

La elaboración del plan de pruebas deberá evaluarse por parte de los docentes en el laboratorio y brindar una realimentación a los estudiantes, de tal forma que los estudiantes presenten un plan de pruebas con todos sus elementos considerando las pruebas de unidad, integración, validación y de sistema.

Ambiente de Aprendizaje: Laboratorio.





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Computadora personal</li> <li>• Software de desarrollo</li> <li>• Bibliotecas para pruebas unitarias</li> </ul> <p><b>Recursos Didácticos:</b></p> <ul style="list-style-type: none"> <li>• Formato de plan de pruebas</li> </ul>	<p>Plan de pruebas</p>	<p><b>Instrumento de Evaluación:</b></p> <ul style="list-style-type: none"> <li>• Guía de observación</li> </ul> <p><b>Criterios de evaluación forma:</b></p> <ul style="list-style-type: none"> <li>• Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>• Respeta la estructura del documento propuesto por el docente</li> <li>• Entrega en tiempo y forma</li> <li>• Redacta sin errores de sintaxis ni de ortografía.</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Establece un propósito acorde al contexto de la aplicación del software.</li> <li>• Define un alcance viable a los tiempos de trabajo.</li> <li>• Incluye un glosario de términos</li> <li>• Considera todos los tipos de las pruebas a aplicar.</li> <li>• Reseña los criterios para evaluar la prueba a aplicar.</li> <li>• Menciona los requerimientos de ambiente de las pruebas.</li> <li>• Indica la calendarización de aplicación y evaluación de las pruebas.</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

<b>Nombre de la Práctica:</b>	Casos de pruebas	<b>N° de la Práctica:</b>	4	<b>Tiempo:</b>	6 horas
<b>Unidad de Competencia:</b>	Desarrolla un plan de pruebas a través de casos de prueba e indicadores para encontrar errores fallos y defectos del producto de software.				
<b>Aprendizajes Esperados Relacionados con la Práctica:</b>	Construye los casos de prueba mediante formatos e indicadores específicos acorde al tipo de prueba para encontrar errores, fallos o defectos.				

**Contenidos de Aprendizaje Relacionados con la Práctica**

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> <li>Elaboración de casos de prueba                             <ul style="list-style-type: none"> <li>Propósito del caso de prueba</li> <li>Prerrequisitos</li> <li>indicadores</li> <li>Valores de entrada</li> <li>Pasos de ejecución</li> <li>Resultado esperado</li> <li>Resultado obtenido</li> <li>Estado de la prueba</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Realiza casos de prueba para cada tipo de prueba, con la finalidad de encontrar errores, fallos y defectos.</li> <li>Práctica 4. Casos de prueba</li> </ul>	<ul style="list-style-type: none"> <li>Trabaja colaborativamente y de forma responsable</li> <li>Muestra pensamiento crítico y analítico</li> <li>Resuelve problemas</li> </ul>

**Estrategia Didáctica y Ambiente de Aprendizaje**

**Estrategia Didáctica:** Design Thinking

El propósito de la práctica es establecer un conjunto de casos de pruebas que den respuesta a las pruebas de unidad, integración, validación y de sistema las cuales se dieron a conocer en la unidad 1 y que se llevarán a cabo para una aplicación de software, preferentemente la que está relacionada con la unidad de aprendizaje Laboratorio de desarrollo de software III.

Los estudiantes, en equipo elaboran uno a uno los casos para las pruebas de unidad, integración, validación y de sistema, considerando los siguientes elementos:

- Propósito del caso de prueba
- Prerrequisitos
- indicadores
- Valores de entrada
- Pasos de ejecución
- Resultado esperado
- Resultado obtenido
- Estado de la prueba

Para la elaboración de cada caso de prueba debe seguir los siguientes pasos:

1. Empatizar: observar, comprender y realizar algunas preguntas sobre el contexto de cada prueba.
2. Definir: definir los prerrequisitos para la aplicación de la prueba
3. Idear: plantear indicadores, valores de entrada y resultado esperado
4. Prototipar: Establecer los pasos de ejecución para la prueba y ejecutar la prueba
5. Validar: presentar en una exposición grupal los casos de prueba para validar, depurar y mejorar los casos de prueba.





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

Durante la elaboración de los casos de prueba el docente debe realimentar el trabajo realizado por los estudiantes, lo estudiantes integran los casos de prueba como parte del plan de pruebas previamente diseñado.

Ambiente de Aprendizaje: Laboratorio

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Computadora personal</li> <li>• Software de desarrollo</li> <li>• Bibliotecas para pruebas unitarias</li> </ul> <p><b>Recursos Didácticos:</b></p> <ul style="list-style-type: none"> <li>• Formato de casos de prueba de acuerdo con el tipo de prueba</li> </ul>	<p>Reporte de los casos de prueba</p>	<p><b>Instrumento de Evaluación:</b> Lista de cotejo</p> <p><b>Criterios de evaluación forma:</b></p> <ul style="list-style-type: none"> <li>• Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>• Respeto la estructura de los casos de prueba plan de pruebas propuesta por el docente</li> <li>• Entrega en tiempo y forma</li> <li>• Redacta sin errores de sintaxis ni de ortografía.</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Crea casos de prueba para los diferentes tipos de las pruebas, tales como pruebas de unidad, integración, validación, sistema y depuración.</li> <li>• Escribe el propósito correcto en cada caso de prueba.</li> <li>• Establece los prerrequisitos de forma completa.</li> <li>• Integra los indicadores que van a usar.</li> <li>• Escribe los valores de entrada en cada uno de los casos de prueba</li> <li>• Describe los pasos de ejecución de forma ordenada.</li> <li>• Define los resultados esperados.</li> <li>• Incluye un espacio para el resultado de la prueba y el estado de la prueba</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

<b>Nombre de la Práctica:</b>	Aplicación de pruebas	<b>N° de la Práctica:</b>	<b>5</b>	<b>Tiempo:</b>	<b>6 horas</b>
<b>Unidad de Competencia:</b>	Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.				
<b>Aprendizajes Esperados Relacionados con la Práctica:</b>	Pone a prueba aplicaciones de software mediante entornos de prueba, herramientas y técnicas para identificar errores, fallos y defectos.				

Contenidos de Aprendizaje Relacionados con la Práctica		
Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> <li>• Aplicación de pruebas</li> <li>• Herramientas y técnicas de pruebas</li> <li>• Proceso de ejecución de las pruebas unitarias</li> <li>• Proceso de ejecución de las pruebas de integración</li> <li>• Proceso de ejecución de las pruebas funcionalidad</li> <li>• Pruebas de sistema</li> <li>• Proceso de ejecución de las pruebas de rendimiento</li> </ul>	<ul style="list-style-type: none"> <li>• Aplica pruebas de software para identificar errores, fallos y defectos en el mismo con el objetivo de garantizar su calidad y correcto funcionamiento.</li> <li>• Práctica 5. Aplicación de pruebas</li> </ul>	<ul style="list-style-type: none"> <li>• Trabaja colaborativamente.</li> <li>• Muestra pensamiento reflexivo, crítico y analítico</li> <li>• Se comunica efectivamente</li> <li>• Muestra responsabilidad personal y social</li> <li>• Respeta las posturas y los comentarios de sus compañeros</li> </ul>

**Estrategia Didáctica y Ambiente de Aprendizaje**

Estrategia Didáctica: Trabajo Colaborativo

- La práctica tiene como propósito que el estudiante aplique casos de prueba previamente elaborados con el objetivo de encontrar errores lógicos en el código, fallos durante la ejecución y defectos por lo que debe proporcionar al estudiante el entorno de desarrollo y de prueba.
- Los estudiantes dirigidos por el profesor titular y auxiliar, propondrán poner a prueba una aplicación de software, preferentemente la que desarrollan en la unidad de aprendizaje Laboratorio de desarrollo de Software III, mediante la ejecución de los casos de prueba para las siguientes pruebas:
  - •Pruebas unitarias
  - •Pruebas de Integración
  - •Prueba de funcionalidad
  - •Pruebas de sistema
  - •Pruebas de rendimiento
- Los estudiantes respetaran los equipos de trabajo conformados en la unidad de aprendizaje Laboratorio de desarrollo de Software III, o bien algún grupo de trabajo con el que se encuentren trabajando un proyecto de software.
- Los roles que deben tomar en cuenta en esta actividad son el de Project manager y Tester.
- El tiempo para llevar a cabo la práctica dependerá de lo que se haya planeado en el plan de pruebas para cada una de las pruebas.
- Finalmente, los estudiantes elaborarán un reporte de casos de prueba con los resultados obtenidos.

Ambiente de Aprendizaje: Laboratorio.

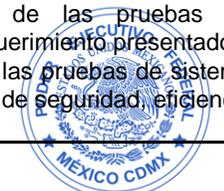




**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Computadora personal</li> <li>• Software de desarrollo</li> <li>• Bibliotecas para pruebas unitarias</li> </ul> <p><b>Recursos Didácticos:</b></p> <ul style="list-style-type: none"> <li>• Formato de casos de pruebas</li> </ul>	<p>Reporte de casos de prueba</p>	<p><b>Instrumento de Evaluación:</b> Guía de observación</p> <p><b>Criterios de evaluación de forma:</b></p> <ul style="list-style-type: none"> <li>• Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>• Entrega en tiempo y forma.</li> <li>• Redacta sin errores de sintaxis ni de ortografía.</li> <li>• Considera un espacio para reportar las pruebas unitarias, integración, funcionalidad, sistema y rendimiento.</li> </ul> <p><b>Criterios de evaluación de forma:</b></p> <ul style="list-style-type: none"> <li>• Realiza un análisis de caminos independientes.</li> <li>• Presenta el grafo de flujo.</li> <li>• Identifica aristas y nodos.</li> <li>• Plantea valores de entrada, esperados de salida para las pruebas unitarias, integración, funcionalidad, sistema y rendimiento.</li> <li>• Descubre errores lógicos.</li> <li>• Comprueba la integración de componentes</li> <li>• Verifica la funcionalidad del software.</li> <li>• Plantea pruebas de sistema.</li> <li>• Presenta los resultados de las pruebas unitarias tomando en cuenta en análisis de caminos independientes.</li> <li>• Interpreta los resultados de las pruebas de integración tomando en cuenta valores de entrada y salida.</li> <li>• Interpreta los resultados de las pruebas de funcionalidad para cada requerimiento presentado.</li> <li>• Interpreta los resultados de las pruebas de sistema tomando en cuenta criterios de seguridad, eficiencia y rendimiento.</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

<b>Nombre de la Práctica:</b>	Calidad del software	<b>N° de la Práctica:</b>	6	<b>Tiempo:</b>	6 horas
<b>Unidad de Competencia:</b>	Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.				
<b>Aprendizajes Esperados Relacionados con la Práctica:</b>	Argumenta los resultados de pruebas obtenidos a través de métricas para lograr identificar las cualidades del software que le proporciona la calidad.				

**Contenidos de Aprendizaje Relacionados con la Práctica**

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> <li>Métricas en las cualidades del software               <ul style="list-style-type: none"> <li>Funcionalidad</li> <li>Confiabilidad</li> <li>Usabilidad</li> <li>Eficiencia</li> <li>Mantenibilidad</li> <li>Portabilidad</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Analiza los resultados de las métricas de calidad para establecer un diagnóstico.</li> <li>Práctica 6. Métricas de calidad en el software</li> </ul>	<ul style="list-style-type: none"> <li>Trabaja colaborativamente.</li> <li>Muestra pensamiento reflexivo, crítico y analítico</li> <li>Se comunica efectivamente</li> <li>Muestra responsabilidad personal y social</li> <li>Respeto las posturas y los comentarios de sus compañeros</li> </ul>

**Estrategia Didáctica y Ambiente de Aprendizaje**

**Estrategia Didáctica:** Trabajo colaborativo

El propósito de la práctica es que el estudiante aplique los conocimientos adquiridos en el aula de clase para analizar resultados de las métricas de calidad para evaluar la calidad del software.

- El docente deberá dar la instrucción a los estudiantes de retomar la aplicación de software que desarrollan como parte de su proyecto en la unidad de aprendizaje Laboratorio de desarrollo de Software III.
- Los estudiantes tendrán que establecer indicadores para cada una de las cualidades del software:
  - ✓ Funcionalidad
  - ✓ Confiabilidad
  - ✓ Usabilidad
  - ✓ Eficiencia
  - ✓ Mantenibilidad
  - ✓ Portabilidad
- Los estudiantes respetaran los equipos de trabajo conformados en la unidad de aprendizaje Laboratorio de desarrollo de Software III, o bien algún grupo de trabajo con el que se encuentren trabajando un proyecto de software.
- Los roles que deben tomar en cuenta en esta actividad son el de Project manager y Tester.
- El tiempo para llevar a cabo la práctica será de seis horas
- Finalmente, él estudiante elaborará un reporte en el que muestre un análisis de los resultados de las métricas de calidad que muestre argumentos para establecer un diagnóstico de la calidad.

Ambiente de Aprendizaje: Laboratorio.





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p><b>Herramientas Tecnológicas:</b></p> <ul style="list-style-type: none"> <li>• Computadora personal</li> <li>• Software de desarrollo</li> </ul> <p><b>Recursos Didácticos:</b> Formatos de métricas de calidad</p>	<p>Reporte de los resultados de las métricas de calidad.</p>	<p><b>Instrumento de evaluación:</b></p> <ul style="list-style-type: none"> <li>• Guía de observación</li> </ul> <p><b>Criterios de evaluación de forma:</b></p> <ul style="list-style-type: none"> <li>• Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>• Entrega en tiempo y forma.</li> <li>• Redacta sin errores de sintaxis ni de ortografía.</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Establece indicadores para las métricas de calidad funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.</li> <li>• Muestra resultados obtenidos</li> <li>• Argumenta los resultados obtenidos</li> <li>• Redacta un diagnóstico con base en los resultados obtenidos para la evaluación de la calidad.</li> </ul>



**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

**PLAN DE EVALUACIÓN SUMATIVA**

N°	Unidad de Competencia	Evidencia Integradora	Criterios e Instrumento de Evaluación	Porcentaje de Acreditación
1	Interpreta la importancia de la calidad en el desarrollo de productos de software mediante el uso de normas y categorías de pruebas para establecer indicadores de las mismas.	Organizador gráfico con la clasificación del tipo de pruebas y las características particulares de cada una.	<p>Instrumento de evaluación</p> <ul style="list-style-type: none"> <li>Lista de cotejo</li> </ul> <p><b>Criterios de evaluación forma:</b></p> <ul style="list-style-type: none"> <li>Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>No tiene errores de ortografía</li> <li>La redacción es clara y mantiene un tono profesional.</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>Identificar de manera correcta los tipos de pruebas.</li> <li>Presenta las características de manera congruente y clara</li> </ul>	30 %
2	Desarrolla un plan de pruebas a través de casos de prueba e indicadores para encontrar errores fallos y defectos del producto de software.	Plan de pruebas con los casos de prueba e indicadores para encontrar errores fallos y defectos del producto de software adecuadas al tipo de prueba.	<p><b>Instrumento de evaluación</b></p> <ul style="list-style-type: none"> <li>Lista de cotejo/Escala de valor</li> </ul> <p><b>Criterios de evaluación de forma:</b></p> <ul style="list-style-type: none"> <li>Incluye carátula: Unidad académica, programa académico, grupo, título, nombres y fecha de entrega.</li> <li>Integra introducción: Describe el contenido, propósito, alcance de las pruebas a su proyecto.</li> <li>El documento tiene el formato proporcionado.</li> </ul> <p><b>Criterios de fondo:</b></p> <ul style="list-style-type: none"> <li>Integra los elementos que incluye un plan de pruebas</li> <li>Las especificaciones son claras y completas.</li> <li>Enlista los tipos de las pruebas a aplicar al proyecto.</li> <li>Reseña los criterios para evaluar la prueba a aplicar.</li> <li>Mencionan los requerimientos de ambiente de las pruebas.</li> <li>Se indica la calendarización de aplicación y evaluación de las pruebas</li> <li>Describe los criterios para evaluar la prueba a aplicar.</li> <li>Los casos de prueba tienen propósito, prerrequisitos, indicadores, valores de entrada, pasos de ejecución, resultado esperado, resultado obtenido, evaluación de las pruebas, y estado de la prueba</li> </ul>	30 %
3			<p><b>Instrumento de evaluación</b></p> <p>Lista de cotejo</p>	40%





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

	<p>Evalúa la calidad del software mediante resultados obtenidos de la aplicación de pruebas para lograr aplicaciones de software aptas para su implementación.</p>	<p>Informe de evaluación de la calidad del software</p>	<p><b>Criterios de forma:</b></p> <ul style="list-style-type: none"> <li>• Integra una caratula con los datos del centro de estudios, nombres de los estudiantes participantes, grupo, carrera y fecha de elaboración.</li> <li>• No tiene errores de ortografía</li> <li>• La redacción es clara y mantiene un tono profesional.</li> </ul> <p><b>Criterios de fondo:</b></p> <ul style="list-style-type: none"> <li>• Establece indicadores en las métricas de calidad.</li> <li>• Muestra los resultados obtenidos de las métricas de calidad.</li> <li>• Analiza e interpreta los resultados obtenidos.</li> </ul>	
<p><b>Propósito de la Unidad de Aprendizaje</b></p>	<p><b>Evidencia Integradora</b></p>	<p><b>Criterios e Instrumento de Evaluación</b></p>	<p><b>Porcentaje de Acreditación</b></p>	
<p>Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.</p>	<p>Informe de la evaluación de la calidad para aplicaciones de software</p>	<p><b>Instrumento de evaluación</b></p> <ul style="list-style-type: none"> <li>• Rúbrica</li> </ul> <p><b>Criterios de evaluación de forma:</b></p> <ul style="list-style-type: none"> <li>• El proceso de las pruebas unitarias, integración, funcionalidad, sistema y rendimiento es de acuerdo al modelo de desarrollo y casos de prueba.</li> <li>• Las herramientas usadas son las especificadas.</li> </ul> <p><b>Criterios de evaluación de fondo:</b></p> <ul style="list-style-type: none"> <li>• Ejecuta correctamente el proceso de las pruebas unitarias, integración, funcionalidad, sistema y rendimiento.</li> <li>• Presenta los resultados de las pruebas unitarias, integración, funcionalidad, sistema y rendimiento.</li> <li>• Interpreta los resultados de las pruebas unitarias, integración, funcionalidad, sistema y rendimiento</li> <li>• Analiza el resultado de las métricas de calidad e indicadores determinando con ello qué calidad tiene el software.</li> </ul>	<p><b>100%</b></p>	





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

**PROGRAMA SINTÉTICO**

PROPÓSITO DE LA UNIDAD DE APRENDIZAJE			
Evalúa la calidad de las aplicaciones de software con responsabilidad social, a través del diseño de casos de prueba, herramientas y técnicas, para determinar su condición y la satisfacción de las necesidades de los usuarios.			
N°	UNIDAD DE COMPETENCIA	APRENDIZAJES ESPERADOS	CONTENIDOS DE APRENDIZAJE/SABERES
1	Interpreta la importancia de la calidad en el desarrollo de productos de software mediante el uso de normas y categorías de pruebas para establecer indicadores de las mismas.	1. Reconoce las métricas en el desarrollo de productos de software reconociendo la importancia de la calidad a través del estudio de error, fallo y defecto.	<p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>• Contextualización de la calidad en el software               <ul style="list-style-type: none"> <li>○ Definición de ingeniería de pruebas</li> <li>○ Calidad en el software</li> <li>○ Error</li> <li>○ Fallo</li> <li>○ Defecto</li> <li>○ Pruebas del software y sus características</li> <li>○ Métricas de revisión</li> <li>○ Revisiones informales</li> <li>○ Revisiones formales</li> <li>○ Aseguramiento de la calidad</li> <li>○ Normas de la calidad (ISO 9126)</li> </ul> </li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Identifica las técnicas de revisión para el aseguramiento de calidad en el software</li> <li>• Práctica 1. Calidad en el software</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Respeta las posturas y los comentarios de sus compañeros</li> <li>• Trabaja colaborativamente</li> <li>• Participa de forma ética y responsable en las actividades durante la clase.</li> </ul>
		2. Identifica los tipos de pruebas mediante el qué prueba y cómo lo prueba, determinando los indicadores de las mismas.	<p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>• Estrategias para las pruebas del software               <ul style="list-style-type: none"> <li>○ Técnica de prueba de caja blanca, camino básico y caja negra</li> <li>○ Proceso de pruebas</li> <li>○ Pruebas de unidad</li> <li>○ Pruebas de integración</li> <li>○ Pruebas de validación</li> <li>○ Pruebas de sistema</li> <li>○ Proceso de depuración</li> </ul> </li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Identifica objetivos e indicadores en las pruebas del software. para justificar la implementación de las pruebas en el software.</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

			<ul style="list-style-type: none"> <li>• Práctica 2. Tipos de pruebas</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Trabaja colaborativamente de forma responsable.</li> <li>• Muestra pensamiento crítico y analítico</li> <li>• Se comunica efectivamente</li> <li>• Respeta las posturas y los comentarios de sus compañeros</li> </ul>
2	<p>Desarrolla un plan de pruebas a través de casos de prueba e indicadores para encontrar errores fallos y defectos del producto de software.</p>	<p>1. Planea las pruebas del software considerando los requerimientos de ambiente de pruebas y un plan de pruebas para establecer un propósito, alcance, un calendario de aplicación, y evaluación.</p>	<p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>• Planificación de las pruebas             <ul style="list-style-type: none"> <li>○ Plan de pruebas</li> <li>○ Propósito y alcance</li> <li>○ Calendarización de aplicación de pruebas</li> <li>○ Calendarización de evaluación</li> <li>○ Requerimientos de ambiente de pruebas</li> <li>○ Glosario de términos</li> <li>○ Validación</li> <li>○ Verificación</li> </ul> </li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Realiza un plan de pruebas orientado a su proyecto para establecer una calendarización de la aplicación de las pruebas.</li> <li>• Práctica 3. Plan de pruebas</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Trabaja colaborativamente de forma responsable.</li> <li>• Muestra pensamiento crítico y analítico</li> <li>• Resuelve problemas</li> </ul>
		<p>2. Construye los casos de prueba mediante formatos e indicadores específicos acorde al tipo de prueba para encontrar errores, fallos o defectos.</p>	<p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>• Elaboración de casos de prueba             <ul style="list-style-type: none"> <li>○ Propósito del plan</li> <li>○ Prerrequisitos</li> <li>○ Indicadores</li> <li>○ Valores de entrada</li> <li>○ Pasos de ejecución</li> <li>○ Resultado esperado</li> <li>○ Resultado obtenido</li> <li>○ Estado de la prueba</li> </ul> </li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Realiza casos de prueba para cada tipo de prueba, con la finalidad de encontrar errores, fallos y defectos.</li> <li>• Práctica 4. Casos de prueba</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Trabaja colaborativamente y de forma responsable</li> <li>• Muestra pensamiento crítico y analítico</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

3	<p>Evalúa la calidad del software mediante resultados obtenidos de la aplicación de pruebas para lograr aplicaciones de software aptas para su implementación.</p>	<p>1. Pone a prueba aplicaciones de software mediante entornos de prueba, herramientas y técnicas para identificar errores, fallos y defectos.</p>	<ul style="list-style-type: none"> <li>• Resuelve problemas</li> </ul> <p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>• Aplicación de pruebas               <ul style="list-style-type: none"> <li>○ Herramientas y técnicas de pruebas</li> <li>○ Proceso de ejecución de las pruebas unitarias</li> <li>○ Proceso de ejecución de las pruebas de integración</li> <li>○ Proceso de ejecución de las pruebas funcionalidad</li> <li>○ Pruebas de Sistema</li> <li>○ Proceso de ejecución de las pruebas de rendimiento</li> </ul> </li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Aplica pruebas de software para identificar errores, fallos y defectos en el mismo con el objetivo de garantizar su calidad y correcto funcionamiento.</li> <li>• Práctica 5. Aplicación de pruebas</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Trabaja colaborativamente.</li> <li>• Muestra pensamiento reflexivo, crítico y analítico</li> <li>• Se comunica efectivamente</li> <li>• Muestra responsabilidad personal y social</li> <li>• Respeta las posturas y los comentarios de sus compañeros</li> </ul>
		<p>2. Argumenta los resultados de pruebas obtenidos a través de métricas para lograr identificar las cualidades del software que le proporciona la calidad.</p>	<p><b>Conceptual:</b></p> <ul style="list-style-type: none"> <li>• Métricas en las cualidades del software               <ul style="list-style-type: none"> <li>○ Funcionalidad</li> <li>○ Confiabilidad</li> <li>○ Usabilidad</li> <li>○ Eficiencia</li> <li>○ Mantenibilidad</li> <li>○ Portabilidad</li> </ul> </li> </ul> <p><b>Procedimental:</b></p> <ul style="list-style-type: none"> <li>• Distingue los indicadores de la calidad del software para establecer valores de entrada y esperados.</li> <li>• Analiza los resultados de las métricas de calidad para establecer un diagnóstico.</li> <li>• Práctica 6. Métricas de calidad en el software</li> </ul> <p><b>Actitudinal:</b></p> <ul style="list-style-type: none"> <li>• Trabaja colaborativamente.</li> <li>• Muestra pensamiento reflexivo, crítico y analítico</li> <li>• Se comunica efectivamente</li> <li>• Muestra responsabilidad personal y social</li> <li>• Respeta las posturas y los comentarios de sus compañeros</li> </ul>





**Programa Académico: Técnico en Programación**

**Unidad de Aprendizaje: Pruebas de Software**

**BIBLIOGRAFÍA BÁSICA Y COMPLEMENTARIA**

Número y Nombre de la Unidad Didáctica	FORMATO APA	CLASIFICACIÓN	
		Básico	Consulta
Unidad 1. Fundamento de las pruebas del software. Unidad 2. Planificación y diseño de las pruebas Unidad 3. Ejecución de pruebas y evaluación de la calidad.	Pressman Roger S, Ph.D. (2010), Ingeniería del software, University of Connecticut, séptima edición.	X	
Unidad 1. Fundamento de las pruebas del software.	A Craftsman’s Approach (2022) Software Testing, Auerbach Publications, Fifth Edition.	X	
Unidad 3. Ejecución de pruebas y evaluación de la calidad.	Graham D. (2018) Foundations of Software Testing: ISTQB Certification, Cengage Learning EMEA, 4th edition.		X
Unidad 3. Ejecución de pruebas y evaluación de la calidad.	Soto A. (2018) Testing Java Microservices, Manning, First Edition.		X
Unidad 3. Ejecución de pruebas y evaluación de la calidad.	Kaczanowski T.(2013) Practical Unit Testing with JUnit and Mockito, Tomek Kaczanowski		X
Unidad 1. Fundamento de las pruebas del software. Unidad 2. Planificación y diseño de las pruebas	Arreola R.(2018) Cómo ser un Tester: Introducción a las Pruebas de Software, Edición Kindle	X	
	Siqueria G. (2018) Ingeniería de Requisitos: Software Orientado al Negocio , Edición Kindle		
Unidad 3. Ejecución de pruebas y evaluación de la calidad.	Lastra L. (2021) Frontend Testing. Edición Kindle	X	
Unidad 2. Planificación y diseño de las pruebas Unidad 3. Ejecución de pruebas y evaluación de la calidad.	Bolaños D. (2008) Pruebas de Software y Junit, un análisis en profundidad y ejemplos prácticos. Pearson	X	
Unidad 3. Ejecución de pruebas y evaluación de la calidad.	Westfall L. (2017) The Certified Software Quality Engineer Handbook, Second Edition, ASQ Quality Press		

