



Programa de Estudios de la Unidad de Aprendizaje:										LABORATORIO DE DESARROLLO DE SOFTWARE IV																			
Clave:	6FP-FM1315				Créditos:	4.50				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	<input type="checkbox"/>		Ciencias Médico Biológicas	<input type="checkbox"/>		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	<input type="checkbox"/>		Científica, Humanística y Tecnológica Básica	<input type="checkbox"/>		Profesional	<input checked="" type="checkbox"/>		Global: <u>72</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	<input type="checkbox"/>		Laboratorio	<input checked="" type="checkbox"/>	Otros ambientes de Aprendizaje	<input type="checkbox"/>		Taller: <u>-</u> Hrs/Semana Total: <u>-</u> Hrs/Semestre																			
Modalidad										Laboratorio: <u>3</u> Hrs/Semana Total: <u>54</u> Hrs/Semestre																			
Escolarizada	<input checked="" type="checkbox"/>	No Escolarizada	<input type="checkbox"/>		Mixta	<input type="checkbox"/>		Otros ambientes de aprendizaje: <u>-</u> Hrs/Semana Total: <u>-</u> Hrs/Semestre																					
Vigencia a Partir:	ENERO 2025										Organización																		
Proceso de Diseño y Autorización:										Por Unidad de Aprendizaje: <input checked="" type="checkbox"/>					Por Área: <input type="checkbox"/>					Por Módulo: <input type="checkbox"/>									
Firma y Sello de Autorización:										 <p>M. EN E.N.A. MARÍA ISABEL ROJAS RUÍZ Directora de Educación Media Superior</p>																			
Elaborado por:	REP. ACAD. NMS				Fecha de Elaboración:	05	09	2024																					
Revisado por:	DEMS				Fecha de Revisión:	25	11	2024																					
Aprobado por:	CTCE-NMS				Fecha de Aprobación:	09	12	2024																					
Autorizado por:	CPA-CGC				Fecha de Autorización:	13	12	2024																					

Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

FUNDAMENTACIÓN

La Unidad de Aprendizaje Laboratorio de Desarrollo de Software IV pertenece al área de formación profesional del Bachillerato Tecnológico Bivalente del Nivel Medio Superior del Instituto Politécnico Nacional, se ubica en el sexto nivel del Plan de Estudios del Programa Académico Técnico en Programación y se imparte en la modalidad escolarizada, de manera obligatoria en la rama del conocimiento de Ingeniería y Ciencias Físico-Matemáticas, desarrollando habilidades en el estudiante requeridas en la industria actual.

Laboratorio de Desarrollo de Software IV, es una unidad de aprendizaje coadyuva a la formación de técnicos en programación capaces de desarrollar proyectos de software, incluyendo su implementación para la automatización de procesos diversos necesarios en industria del Software. Esta unidad de aprendizaje está fundamentada en el Modelo Educativo Institucional vigente y en la educación para la industria por ello se empleará la metodología didáctica de aprendizaje basado en proyectos, misma con la que se busca poner en práctica los conocimientos teórico-prácticos adquiridos en las unidades de aprendizaje correspondientes al sexto nivel de la Carrera Técnico en Programación (Ciberseguridad, Introducción al Análisis de Datos, Fundamentos de Inteligencia Artificial), así como los adquiridos en las unidades de aprendizaje del área profesional correspondientes al tercer, cuarto y quinto nivel.

El Laboratorio de Desarrollo de Software IV promueve el aprendizaje práctico y experiencial, mediante la ejecución de proyectos de software que responden a las necesidades reales del sector. Los estudiantes desarrollarán habilidades de análisis y aplicación del plan de mejora continua, planeación estratégica y las etapas de la reingeniería de proyectos de software, aplicando conocimientos teóricos-prácticos en un entorno simulado a las condiciones laborales.

Por lo anterior, el docente que imparta Laboratorio de Desarrollo de Software IV debe dominar las habilidades y los conocimientos a desarrollar en el estudiante, de manera que pueda contribuir a su formación integral con las competencias esenciales para la industria, 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 y flexible.

El aprendizaje se centra en el estudiante, es decir es protagonista activo de su propio proceso de aprendizaje, siendo activo, participativo y responsable. Se espera que desarrolle habilidades blandas como pensamiento crítico, creatividad, resolución de conflictos, habilidades comunicativas, gestión de tiempos y trabajo colaborativo, en conjunto de una aplicación de sus conocimientos en proyectos reales. El enfoque educativo promueve la autonomía, la autogestión y la adaptación a los cambios constantes del entorno, preparando a los estudiantes para enfrentar los desafíos del mundo actual.

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 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. 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.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

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 Laboratorio de Desarrollo de Software III, 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.

La dinámica de trabajo en el laboratorio consta de la conformación de varios equipos de trabajo los cuales desarrollan proyectos de software con herramientas de programación adecuadas a la problemática a automatizar. Por esta situación se debe **un docente titular y dos docentes auxiliares**, que estarán reforzando las actividades dentro de los laboratorios. Los productos y desempeños que desarrolle el estudiante durante el desarrollo del semestre serán integradas en un portafolio de evidencias de aprendizaje y las actividades que se trabaje en equipo se registrarán en un portafolio colaborativo. Los portafolios de evidencias contendrán las evaluaciones correspondientes de los cuestionarios, ejercicios, programas, de cada unidad en forma digital, para facilitar su manejo. La importancia de contar con dicha cantidad de docentes en la actividad del 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 la integridad física del estudiante, el correcto empleo de los equipos de cómputo y las herramientas, en las áreas relacionadas con los laboratorios o talleres del Programa Académico de Técnico en Programación.



DESCRIPCIÓN DE LA UNIDAD DE APRENDIZAJE

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Propósito de la Unidad de Aprendizaje

Maximiza proyectos de software en sus funcionalidades ajustándolos a las necesidades del negocio a través de la mejora continua, reingeniería de software, las competencias técnicas desarrolladas en las unidades de aprendizaje del área profesional y del trabajo colaborativo, para su puesta en producción.

Unidad 1: Optimización de Proyectos

Unidad de competencia	Aprendizajes esperados	Contenidos de aprendizaje
<p>Evalúa el proyecto de software existente, a través del análisis exhaustivo de las áreas críticas de mejora y escalabilidad de procesos del negocio estableciendo el plan de desarrollo de software, empleando un pensamiento crítico para las mejoras que garanticen la eficiencia, eficacia y sostenibilidad del software con la finalidad de optimizar la toma de decisiones.</p>	<p>Identifica áreas críticas de mejora, a través de un análisis exhaustivo en el proyecto de software existente para la optimización de procesos y escalabilidad de los mismos.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> • Recopilación de requerimientos del sistema. • Evaluación de eficiencia y eficacia de los requerimientos. • Revisión y priorización de requerimientos. • Redefinición de requerimientos. <p>Procedimental:</p> <ul style="list-style-type: none"> • Aplica técnicas de levantamiento de requerimientos de software con los usuarios finales para asegurar que el software cumpla con sus necesidades. • Redacta documentos de requisitos detallados y estructurados para asegurar la comprensión y ejecución exitosa de un proyecto de software. • Realiza pruebas de aceptación rigurosas para para garantizar la conformidad de los nuevos requisitos. <p>Actitudinal:</p> <ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Analiza información de manera crítica. • Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad. • Se comunica asertivamente y respetuosa. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

	<p>Estructura el plan de desarrollo de proyectos de software, incorporando cronogramas detallados y estrategias de mitigación de riesgos para implementar las mejoras y nuevos requisitos.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> • Gestión del cambio de requerimientos. • Planificación de las tareas del desarrollo de software. • Gráfica de Gantt y ruta crítica. <p>Procedimental:</p> <ul style="list-style-type: none"> • Elabora la planificación de las tareas del desarrollo del software para responder a los cambios y garantizar la calidad del sistema. • Evalúa lenguajes de programación, frameworks, API's, hardware, software, redes y plataformas utilizadas apropiados para optimizar el rendimiento y la escalabilidad del proyecto del software. <p>Actitudinal:</p> <ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Analiza información de manera crítica. • Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad. • Se comunica asertivamente y respetuosa. • Responde de manera adecuada a las emociones de sus compañeros. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido.
<p>Unidad 2: Restructuración de Proyectos</p>		
<p>Unidad de competencia</p>	<p>Aprendizajes esperados</p>	<p>Contenidos de aprendizaje</p>
<p>Mejora el proyecto de software a través de una reestructuración de la arquitectura, la refactorización del código y la integración de tecnologías emergentes, con el fin de optimizar el rendimiento, escalabilidad, mantenibilidad y capacidad de adaptación, facilitando así la toma de decisiones basadas en análisis estadísticos y proyecciones con la implementación de la inteligencia artificial.</p>	<p>Emplea el lenguaje de modelado unificado (UML), en el rediseño de interacciones de los usuarios con el sistema, la arquitectura del software y el flujo de los procesos para mejorar su eficiencia, escalabilidad y sostenibilidad.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> • Metodología de desarrollo ágil. • Patrones de Diseño. • Arquitecturas de software. • Lenguaje de modelado unificado para el diseño del software. • Herramientas de diagramación UML. • Modelado de la base de datos relacional o no relacional del proyecto de software. • Normalización de bases de datos relacionales. <p>Procedimental:</p>





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

		<ul style="list-style-type: none"> • Modela el diseño, arquitectura, flujos, componentes y despliegue del proyecto de software web conforme al estándar UML para garantizar una representación precisa y comprensible del sistema. • Diseña interfaces gráficas web intuitivas y visualmente atractivas para optimizar la experiencia del usuario. • Diseña la interfaz gráfica de la aplicación móvil que facilite la interacción del usuario con las funcionalidades del sistema en dispositivos portátiles. • Diseña bases de datos relacionales utilizando las formas normales con el objetivo de minimizar la redundancia y maximizar la consistencia de los datos del proyecto. • Diseña bases de datos no relacionales que se adapten las necesidades cambiantes de los datos para garantizar la escalabilidad y la eficiencia del sistema. <p>Actitudinal:</p> <ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Desarrolla soluciones creativas. • Analiza información de manera crítica. • Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad. • Se comunica asertivamente y respetuosa. • Responde de manera adecuada a las emociones de sus compañeros. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido.
	<p>Establece nuevas funcionalidades y tecnologías, en el proyecto de software, a través de una refactorización del código, para mejorar su rendimiento, funcionalidad, calidad, modularidad, seguridad y mantenibilidad.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> • Lenguaje de programación. • Lenguaje de programación móvil. • Entorno de Desarrollo Integrado. • Frameworks. • Entornos de ejecución del lado del servidor.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

		<ul style="list-style-type: none"> • Conceptos estadísticos, distribuciones, regresiones, y análisis descriptivo. • Modelos Predictivos. • Criptografía y autenticación. • Fundamentos de IA. <p>Procedimental:</p> <ul style="list-style-type: none"> • Optimiza la codificación del proyecto mediante el uso de un IDE para aumentar la productividad y la calidad del código. • Emplea frameworks para acelerar el desarrollo y garantizar la escalabilidad de proyectos. • Reescribe el código para el proyecto de software mediante un lenguaje de programación web para la solución a la problemática planteada. • Emplea un lenguaje de programación móvil específico, adaptado a las necesidades del proyecto, para desarrollar una aplicación móvil funcional y eficiente. • Conecta una aplicación a una base de datos (relacional o no relacional) con el fin de almacenar y recuperar datos. • Emplea técnicas estadísticas y de modelado predictivo en los datos para tomar decisiones y realizar pronósticos. • Aplica técnicas de encriptación, autenticación, y control de acceso en el software con el fin de garantizar la seguridad del software. • Integra APIs de IA de manera eficiente para para mejorar la funcionalidad y la experiencia del usuario. <p>Actitudinal:</p> <ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Desarrolla soluciones creativas. • Analiza información de manera crítica. • Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad.
--	--	---





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

		<ul style="list-style-type: none"> • Se comunica asertivamente y respetuosa. • Responde de manera adecuada a las emociones de sus compañeros. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido.
Unidad 3: Implementación de Proyectos		
Unidad de competencia	Aprendizajes esperados	Contenidos de aprendizaje
<p>Planea la entrega de un proyecto de software cumpliendo con los estándares de calidad y los requisitos técnicos y del negocio, a través de la ejecución exhaustiva de casos de prueba para validar sus funcionalidades, gestionando situaciones complejas con eficiencia en la adecuación de un sistema de información en un entorno de producción configurado que permita un rendimiento óptimo, acceso eficiente y seguro por parte de los usuarios.</p>	<p>Testa el proyecto de software con el fin de identificar y corregir errores antes de la puesta en producción, validando las funcionalidades requeridas y asegurando que el sistema de información cumpla con los estándares de calidad establecidos.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> • Técnicas de pruebas. • Herramientas para la automatización de pruebas. <p>Procedimental:</p> <ul style="list-style-type: none"> • Ejecuta pruebas manuales sobre el proyecto de software para identificar defectos y asegurar la calidad del proyecto de software. • Utiliza herramientas de automatización de pruebas sobre el proyecto de software implementando scripts de prueba para garantizar la calidad del software y detectar defectos. <p>Actitudinal:</p> <ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Analiza información de manera crítica. • Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad. • Se comunica asertivamente y respetuosa. • Responde de manera adecuada a las emociones de sus compañeros. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido.
	<p>Adapta el proyecto de software en un entorno de producción configurado correctamente para su funcionamiento óptimo conforme a los requisitos técnicos y del negocio para el acceso de los usuarios.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> • Fundamentos de Despliegue de Software. • Arquitectura de Aplicaciones. • Seguridad en el Despliegue. <p>Procedimental:</p>



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

		<ul style="list-style-type: none"> • Configura servidores, bases de datos y recursos en la nube utilizando herramientas de automatización para agilizar el despliegue y reducir errores de configuración. • Configura variables de entorno, rutas de acceso, configuraciones de red, y otros parámetros necesarios para el funcionamiento del proyecto de software en el despliegue. • Usa de herramientas para automatizar la compilación y empaquetado del código. • Configura los mecanismos de seguridad en el entorno de despliegue para prevenir ataques cibernéticos comunes y proteger los datos sensibles de la aplicación. <p>Actitudinal:</p> <ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Reconoce, entiende y gestiona las propias emociones y las de los integrantes de su equipo de manera efectiva. • Analiza información de manera crítica. Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad. • Se comunica asertivamente y respetuosa. • Responde de manera adecuada a las emociones de sus compañeros. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido
--	--	---





➔ MATRIZ DE VINCULACIÓN ←

COMPETENCIAS PARA EL SIGLO XXI HABILIDADES BLANDAS Y SOCIOEMOCIONALES	Unidad de Competencia 1		Unidad de Competencia 2		Unidad de Competencia 3	
	AE 1	AE 2	AE 1	AE 2	AE 1	AE 2
Trabajo en equipo	X	X	X	X	X	X
Creatividad			X	X		
Innovación			X	X		
Inteligencia emocional			X	X		X
Pensamiento crítico	X	X	X	X	X	X
Liderazgo	X	X	X	X	X	X
Adaptabilidad	X	X	X	X	X	X
Proactividad	X	X	X	X	X	X
Empatía	X	X	X	X	X	X
Comunicación asertiva	X	X	X	X	X	X
Apropiación de las tecnologías digitales	X	X	X	X	X	X
Responsabilidad personal y social	X	X	X	X	X	X



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

PERFIL DOCENTE

El docente que imparta la Unidad de Aprendizaje de **Laboratorio de Desarrollo de Software IV** 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 de habilidades y competencias a partir del uso de nuevas tecnologías.

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.
- Emplea habilidades digitales actualizadas para el desarrollo de la Unidad de Aprendizaje.
- Adaptación a las nuevas tendencias y tecnologías emergentes en el campo del desarrollo de software.

En el campo pedagógico:

- Fomentar procesos de enseñanza que le permitan interpretar y resolver las necesidades de aprendizaje del estudiante, tomando en cuenta sus capacidades, habilidades, estilos de aprendizaje, 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.
- Incentivar el desarrollo de competencias blandas en el estudiante, enfatizando en el pensamiento crítico, la resolución de problemas, comunicación efectiva, creatividad, innovación y la adaptación a nuevas tecnologías.

En el campo de la investigación:

- Fortalecer el trabajo académico a partir del aprovechamiento de los resultados y productos de los proyectos de investigación.
- Explorar contenidos relevantes y novedosos en el campo del desarrollo de software.

Perfil Profesional

- Licenciatura preferentemente en: Ingeniería en Sistemas Computacionales, Ingeniería en Inteligencia Artificial, Ciencia de Datos, Computación; sin descartar Licenciatura en: Informática, Ingeniería Telemática, Ingeniería en Comunicaciones y Electrónica o afines. Se valorará especialmente al docente que cuente con título de licenciatura, antes mencionada, y tenga Maestría en Ciencias de la Computación, en Gestión de la Innovación en Tecnologías de la Información y Comunicación, en Educación o afín; preponderantemente con experiencia de al menos dos años en el área docente.
- Preferentemente contar con al menos dos años en la iniciativa pública o privada aplicando los conocimientos de la unidad de aprendizaje.

Para la unidad de aprendizaje de **Laboratorio de Desarrollo de Software IV**, se requiere de **un docente titular y dos docentes auxiliares, preferentemente cubriendo el perfil anteriormente descrito** para alcanzar las unidades de competencia de este programa académico mediante la impartición de cátedra, diseñando estrategias didácticas y ejecutando técnicas y tecnologías para el desarrollo de proyectos de software, facilitando, orientando y guiando a los estudiantes en la aplicación de las técnicas y tecnologías en el desarrollo de sus proyectos de software, además en realizar y evaluar las etapas del desarrollo del proyecto de software.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

ESTRUCTURA DIDÁCTICA

Unidad Didáctica 1:	Optimización de Proyectos	Nivel:	Sexto
Propósito General:	Maximiza proyectos de software en sus funcionalidades ajustándolos a las necesidades del negocio a través de la mejora continua, reingeniería de software, las competencias técnicas desarrolladas en las unidades de aprendizaje del área profesional y del trabajo colaborativo, para su puesta en producción.		
Unidad de Competencia No 1:	Evalúa el proyecto de software existente, a través del análisis exhaustivo de las áreas críticas de mejora y escalabilidad de procesos del negocio estableciendo el plan de desarrollo de software, empleando un pensamiento crítico para las mejoras que garanticen la eficiencia, eficacia y sostenibilidad del software con la finalidad de optimizar la toma de decisiones.		
Aprendizaje Esperado No 1:	Identifica áreas críticas de mejora, a través de un análisis exhaustivo en el proyecto de software existente para la optimización de procesos y escalabilidad de los mismos.	Tiempo estimado para obtener el Aprendizaje Esperado:	8 horas.
Contenidos de Aprendizaje			
Conceptuales	Procedimentales	Actitudinales	
<ul style="list-style-type: none"> Recopilación de requerimientos del sistema. Evaluación de eficiencia y eficacia de los requerimientos. Revisión y priorización de requerimientos. Redefinición de requerimientos. 	<ul style="list-style-type: none"> Aplica técnicas de levantamiento de requerimientos de software con los usuarios finales para asegurar que el software cumpla con sus necesidades. Redacta documentos de requisitos detallados y estructurados para asegurar la comprensión y ejecución exitosa de un proyecto de software. Realiza pruebas de aceptación rigurosas para garantizar la conformidad de los nuevos requisitos. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido. 	
Estrategia Didáctica y Ambiente de Aprendizaje			
<p>Estrategia Didáctica: Se usa una metodología de Aprendizaje Orientado a Proyectos (AOP).</p> <p>Fase 1 de la metodología Aprendizaje Orientado a Proyectos: Redefinición y priorización de los objetivos del proyecto, los entregables, el impacto esperado. Se pueden presentar opciones de mejora del proyecto existente.</p> <p>Evaluación Diagnóstica, antes de utilizar la metodología de Aprendizaje Orientado a Proyectos (AOP) se realiza una exploración previa de conocimientos adquiridos en la unidad de aprendizaje de LDS III para identificar las áreas de oportunidad de los estudiantes y a partir de ello poder iniciar con los nuevos conocimientos que se les darán y presentarán.</p>			



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

- **Apertura:** El docente, hace una exploración de conocimientos general, con una lluvia de ideas y conceptos básicos sobre los métodos de análisis para la administración de proyectos de software, así como la viabilidad de proyectos de software, con esta exploración general el docente busca identificar los conocimientos previos que el estudiante tenga sobre dichos temas.
- **Desarrollo:** El docente hace una presentación electrónica y explica cómo se trabajará mediante las fases de la metodología del Aprendizaje Orientado a Proyectos, así como las metodologías más importantes para la gestión de proyectos y otra más para la viabilidad de los proyectos. El docente solicita al estudiante que se conserve el equipo de trabajo colaborativo establecido en la unidad de aprendizaje LDS III, con el fin de dar continuidad al desarrollo del proyecto.
- El docente junto con el estudiante evalúa la eficiencia y eficacia de los requerimientos del proyecto, a su vez que los priorizan y los redefinen con el fin de establecer un software de calidad.
- **Cierre:** El estudiante se reúne en equipo de trabajo colaborativo para iniciar con el análisis de la eficiencia y eficacia de los requerimientos del proyecto y redefinirlos mediante la utilización de técnicas, el docente responde las preguntas o dudas que el estudiante tuviera para facilitar el aprendizaje orientado a proyectos.

Ambiente de Aprendizaje:

Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, para el desarrollo de la práctica, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> • Equipo de Cómputo (PC o Lap Top) • Procesadores de texto • Herramientas de software digitales adecuadas para realizar las actividades solicitadas. • Un entorno de desarrollo integrado (IDE) • Conexión a internet • Video proyector • Aula Virtual <p>Plataformas:</p> <ul style="list-style-type: none"> • Software orientado a la unidad didáctica desarrollada, ejemplo Google Classroom o Microsoft Teams. <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Libros digitales o impresos relacionados al tema. • Infografías y Videos relacionados al tema. 	<p>Práctica 1: Redefinición de requerimientos.</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Rúbrica. <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. • Entrega de archivos electrónicos de la redefinición de requerimientos en un documento pdf en laboratorio y/o plataforma virtual. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Aplica un análisis detallado de los requerimientos actuales del proyecto, identificando las fortalezas y debilidades de los requerimientos existentes.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

<ul style="list-style-type: none"> • Presentaciones electrónicas relacionadas al tema. • Formularios interactivos relacionados al tema. • Páginas, sitios web y aplicaciones móviles relacionadas al tema. 		<ul style="list-style-type: none"> • Demuestra un conocimiento sólido de al menos dos técnicas de priorización de requerimientos (por ejemplo, MoSCoW, Kano, costo-beneficio). • Aplica la técnica seleccionada de manera adecuada al proyecto de software, identificando correctamente los criterios de priorización. • Explica de forma clara y concisa los motivos de la priorización asignada a cada requerimiento, respaldando sus decisiones con argumentos sólidos. • Identifica correctamente los requerimientos que necesitan ser redefinidos debido a ambigüedades, inconsistencias o cambios en el contexto del proyecto de software. • Sugiere modificaciones a los requerimientos que los hagan más claros, precisos y completos, sin comprometer los objetivos del proyecto de software. • Evalúa el impacto de las modificaciones propuestas en otras partes del proyecto y en los costos y plazos.
---	--	--



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Unidad Didáctica 1:	Optimización de Proyectos	Nivel:	Sexto
Propósito General:	Maximizar proyectos de software en sus funcionalidades ajustándolos a las necesidades del negocio a través de la mejora continua, reingeniería de software, las competencias técnicas desarrolladas en las unidades de aprendizaje del área profesional y del trabajo colaborativo, para su puesta en producción.		
Unidad de Competencia No 1:	Evalúa el proyecto de software existente, a través del análisis exhaustivo de las áreas críticas de mejora y escalabilidad de procesos del negocio estableciendo el plan de desarrollo de software, empleando un pensamiento crítico para las mejoras que garanticen la eficiencia, eficacia y sostenibilidad del software con la finalidad de optimizar la toma de decisiones.		
Aprendizaje Esperado No 2:	Estructura el plan de desarrollo de proyectos de software, incorporando cronogramas detallados y estrategias de mitigación de riesgos para implementar las mejoras y nuevos requisitos.	Tiempo estimado para obtener el Aprendizaje Esperado:	8 horas.

Contenidos de Aprendizaje

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> Gestión del cambio de requerimientos. Planificación de las tareas del desarrollo de software. Gráfica de Gantt y ruta crítica. 	<ul style="list-style-type: none"> Elabora la planificación de las tareas del desarrollo del software para responder a los cambios y garantizar la calidad del sistema. Evalúa lenguajes de programación, frameworks, API's, hardware, software, redes y plataformas utilizadas apropiados para optimizar el rendimiento y la escalabilidad del proyecto del software. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Responde de manera adecuada a las emociones de sus compañeros. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia Didáctica: Se usa una metodología de Aprendizaje Orientado a Proyectos (AOP).

Fase 2 de la metodología Aprendizaje Orientado a Proyectos: Exploración, búsqueda, tratamiento y análisis de la información para la modificación, mejora y escalabilidad de los requerimientos, en esta fase también se redefinen el cronograma de actividades y la ruta crítica.

Evaluación Formativa, se realiza una evaluación intermedia o continua para identificar las competencias que han adquirido los estudiantes hasta el momento del primer momento de evaluación.

- Apertura:** El docente presenta de manera digital un ejemplo con un análisis de un plan de gestión de cambios de requerimientos, ruta crítica y diagrama de Gantt, a manera de que el estudiante, construya y conceptualice su propio conocimiento mediante estos ejemplos.



Programa Académico: Técnico en Programación

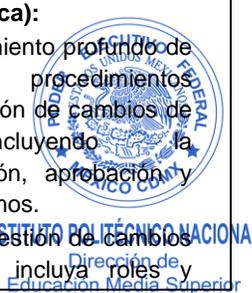
Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

- **Desarrollo:** El estudiante se reúne por equipos para realizar en base a los ejemplos mostrados por el docente, su plan de gestión de cambios, la ruta crítica y el diagrama de Gantt del proyecto de software presentado el cual contiene elementos de tipo web como móvil.
- **Cierre:** El docente supervisa los equipos de trabajo y responde las preguntas o dudas que el estudiante tuviera para facilitar el aprendizaje orientado a proyectos.

Ambiente de Aprendizaje:

Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, para el desarrollo de la práctica, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> • Equipo de Cómputo (PC o Lap Top) • Procesadores de texto • Herramientas de software digitales adecuadas para realizar las actividades solicitadas. • Un entorno de desarrollo integrado (IDE) • Conexión a internet • Video proyector • Aula Virtual <p>Plataformas:</p> <ul style="list-style-type: none"> • Software orientado a la unidad didáctica desarrollada, ejemplo el Google Classroom o Microsoft Teams. <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Libros digitales o impresos relacionados al tema. • Infografías y Videos relacionados al tema. • Presentaciones electrónicas relacionadas al tema. • Formularios interactivos relacionados al tema. • Páginas, sitios web y aplicaciones móviles relacionadas al tema. 	<p>Práctica 2: Gestión del cambio de requerimientos.</p>	<p>Instrumento de evaluación: Rúbrica.</p> <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. • Entrega de archivos electrónicos del documento del plan de Gestión del cambio de requerimientos en un documento pdf en laboratorio y/o plataforma virtual. • Entrega de archivos electrónicos Gráfica de Gantt y ruta crítica en un documento pdf en laboratorio y/o plataforma virtual. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Demuestra un entendimiento profundo de los procesos y procedimientos involucrados en la gestión de cambios de requerimientos, incluyendo la identificación, evaluación, aprobación y seguimiento de los mismos. • Desarrolla un plan de gestión de cambios claro y conciso, que incluya roles y





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

		<p>responsabilidades, procedimientos de solicitud y aprobación, y mecanismos de seguimiento.</p> <ul style="list-style-type: none"> • Construye una matriz de rastreabilidad que permita vincular los requerimientos con otros elementos del proyecto de software y facilitar el seguimiento de los cambios. • Analiza el impacto de los cambios de requerimientos en el cronograma del proyecto de software e identifica las tareas afectadas. • Actualiza el diagrama de Gantt y la ruta crítica de manera precisa para reflejar los cambios realizados. • Propone ajustes en la asignación de recursos para minimizar el impacto de los cambios en el cronograma.
--	--	--



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Propósito General:	Maximizar proyectos de software en sus funcionalidades ajustándolos a las necesidades del negocio a través de la mejora continua, reingeniería de software, las competencias técnicas desarrolladas en las unidades de aprendizaje del área profesional y del trabajo colaborativo, para su puesta en producción.	
Unidad de Competencia No 2:	Mejora el proyecto de software a través de una reestructuración de la arquitectura, la refactorización del código y la integración de tecnologías emergentes, con el fin de optimizar el rendimiento, escalabilidad, mantenibilidad y capacidad de adaptación, facilitando así la toma de decisiones basadas en análisis estadísticos y proyecciones con la implementación de la inteligencia artificial.	
Aprendizaje Esperado No 1:	Emplea el lenguaje de modelado unificado (UML), en el rediseño de interacciones de los usuarios con el sistema, la arquitectura del software y el flujo de los procesos para mejorar su eficiencia, escalabilidad y sostenibilidad.	Tiempo estimado para obtener el Aprendizaje Esperado: 16 horas.

Contenidos de Aprendizaje

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> • Metodología de desarrollo ágil. • Patrones de Diseño. • Arquitecturas de software. • Lenguaje de modelado unificado para el diseño del software. • Herramientas de diagramación UML. • Modelado de la base de datos relacional o no relacional del proyecto de software. • Normalización de bases de datos relacionales. 	<ul style="list-style-type: none"> • Modela el diseño, arquitectura, flujos, componentes y despliegue del proyecto de software web conforme al estándar UML para garantizar una representación precisa y comprensible del sistema. • Diseña interfaces gráficas web intuitivas y visualmente atractivas para optimizar la experiencia del usuario. • Diseña la interfaz gráfica de la aplicación móvil que facilite la interacción del usuario con las funcionalidades del sistema en dispositivos portátiles. • Diseña bases de datos relacionales utilizando las formas normales con el objetivo de minimizar la redundancia y maximizar la consistencia de los datos del proyecto. • Diseña bases de datos no relacionales que se adapten las necesidades cambiantes de los datos para garantizar la escalabilidad y la eficiencia del sistema. 	<ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Desarrolla soluciones creativas. • Analiza información de manera crítica. • Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad. • Se comunica asertivamente y respetuosa. • Responde de manera adecuada a las emociones de sus compañeros. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia Didáctica: Se usa una metodología de Aprendizaje Orientado a Proyectos (AOP).

Fase 3 de la metodología Aprendizaje Orientado a Proyectos: Definición del software a utilizar en el proyecto y elaboración de los diagramas UML, dando seguimiento a implementación de la metodología ágil, se diseñará una estrategia didáctica que permita a los estudiantes analizar y evaluar las necesidades de sus proyectos. A través de sesiones interactivas, los participantes identificarán y definirán los objetivos del proyecto de software, aplicando principios ágiles en la planificación de sprints. Durante las reuniones diarias, se discutirán los avances y se revisarán los desafíos, fomentando un ambiente de colaboración.

Al final de cada sprint, los equipos presentarán sus resultados y reflexionarán sobre el proceso, sintetizando lecciones aprendidas para mejorar continuamente. Esta metodología no solo promoverá el aprendizaje práctico, sino que también desarrollará habilidades críticas para la gestión ágil de proyectos.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Evaluación Formativa, se realiza una evaluación intermedia o continua para identificar las competencias que han adquirido los alumnos hasta el momento del segundo momento de evaluación.

- **Apertura:** EL docente mencionará diferentes frameworks y SGBD para conocimiento general, se realizará un análisis para la selección del software adecuado al proyecto partiendo de la viabilidad del mismo, explica la configuración de los diferentes ambientes de desarrollo, así como la interacción entre los mismo para trabajar de manera colaborativa entre los mismo orientados al proyecto de software.
- **Desarrollo:** Los estudiantes en equipos analizan la viabilidad de sus proyectos, considerando los recursos disponibles, el tiempo y los conocimientos previos, y así seleccionar el framework y el SGBD más adecuados para el proyecto de software y aplicarán principios ágiles para planificar sus sprints, estableciendo tareas, prioridades y entregables; realizando reuniones diarias para discutir los avances, los obstáculos y los planes para las tareas siguientes y al finalizar cada sprint identificar las áreas de mejora y ajustar el proceso de desarrollo del software. Realizará en el proceso el rediseño de la arquitectura, los flujos, los componentes, el diseño de la base de datos y diseño de la interfaz gráfica de la aplicación web y móvil.
- **Cierre:** El docente supervisa los equipos de trabajo y responde dudas que los estudiantes presentan en sus sprints, así como en la elección del framework y SGBD adecuado a su proyecto de software.

Ambiente de Aprendizaje:

Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, para el desarrollo de la práctica, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> • Equipo de Cómputo (PC o Lap Top) • Procesadores de texto • Herramientas de software digitales adecuadas para realizar las actividades solicitadas. • Un entorno de desarrollo integrado (IDE) • Conexión a internet • Video proyector • Aula Virtual <p>Plataformas:</p> <ul style="list-style-type: none"> • Software orientado a la unidad didáctica desarrollada, ejemplo el Classroom o Teams. 	<p>Práctica 3: Ciclo de vida de los datos.</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Rúbrica. <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto • Entrega de archivos electrónicos modelo de reestructuración del proyecto. (usa alguna herramienta de modelado) en un documento pdf en laboratorio y/o plataforma Virtual de Educación Media Superior





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

<p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Libros digitales o impresos relacionados al tema. • Infografías y Videos relacionados al tema. • Presentaciones electrónicas relacionadas al tema. • Formularios interactivos relacionados al tema. • Páginas, sitios web y aplicaciones móviles relacionadas al tema. 		<p>Entrega de actividades en tiempo y forma solicitados.</p> <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Modela la documentación de forma clara, concisa y precisa, utilizando el lenguaje de modelado unificado UML, y evitando ambigüedades. • Refleja el modelo de manera precisa los requerimientos funcionales y no funcionales del sistema, demostrando una clara comprensión de estos. • Propone modelos arquitectónicos y de diseño que optimicen el rendimiento, la escalabilidad y la mantenibilidad del sistema web y móvil. • Diseña la interfaz gráfica intuitiva y con una experiencia de usuario, tanto para la versión web como para la móvil, considerando las características específicas de cada sistema operativo. • interrelaciona de forma lógica y coherente la arquitectura, flujos, componentes, base de datos e interfaz gráfica.
---	--	---

Unidad Didáctica 2:	Reestructuración de Proyectos	Nivel:	Sexto
Propósito General:	Maximizar proyectos de software en sus funcionalidades ajustándolos a las necesidades del negocio a través de la mejora continua, reingeniería de software, las competencias técnicas desarrolladas en las unidades de aprendizaje del área profesional y de trabajo colaborativo, para su puesta en producción.		



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Unidad de Competencia No 2:	Mejora el proyecto de software a través de una reestructuración de la arquitectura, la refactorización del código y la integración de tecnologías emergentes, con el fin de optimizar el rendimiento, escalabilidad, mantenibilidad y capacidad de adaptación, facilitando así la toma de decisiones basadas en análisis estadísticos y proyecciones con la implementación de la inteligencia artificial.		
Aprendizaje Esperado No 2:	Establece nuevas funcionalidades y tecnologías, en el proyecto de software, a través de una refactorización del código, para mejorar su rendimiento, funcionalidad, calidad, modularidad, seguridad y mantenibilidad.	Tiempo estimado para obtener el Aprendizaje Esperado:	16 horas.

Contenidos de Aprendizaje

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> Lenguaje de programación. Lenguaje de programación móvil. Entorno de Desarrollo Integrado. Frameworks. Entornos de ejecución del lado del servidor. Conceptos estadísticos, distribuciones, regresiones, y análisis descriptivo. Modelos Predictivos. Criptografía y autenticación. Fundamentos de IA. 	<ul style="list-style-type: none"> Optimiza la codificación del proyecto mediante el uso de un IDE para aumentar la productividad y la calidad del código. Emplea frameworks para acelerar el desarrollo y garantizar la escalabilidad de proyectos. Reescribe el código para el proyecto de software mediante un lenguaje de programación web para la solución a la problemática planteada. Emplea un lenguaje de programación móvil específico, adaptado a las necesidades del proyecto, para desarrollar una aplicación móvil funcional y eficiente. Conecta una aplicación a una base de datos (relacional o no relacional) con el fin de almacenar y recuperar datos. Emplea técnicas estadísticas y de modelado predictivo en los datos para tomar decisiones y realizar pronósticos. Aplica técnicas de encriptación, autenticación, y control de acceso en el software con el fin de garantizar la seguridad del software. Integra APIs de IA de manera eficiente para para mejorar la funcionalidad y la experiencia del usuario. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Desarrolla soluciones creativas. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Responde de manera adecuada a las emociones de sus compañeros. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia Didáctica: Se usa una metodología de Aprendizaje Orientado a Proyectos (AOP).

Fase 3 de la metodología Aprendizaje Orientado a Proyectos: Revisión y análisis de la información de la problemática a resolver, en esta fase se lleva a cabo el ciclo de vida de los datos lo correspondiente a la fase de planeación y recolección, almacenamiento y manejo y limpieza y procesamiento.

Evaluación Formativa, se realiza una evaluación intermedia o continua para identificar las competencias que han adquirido los alumnos hasta el momento del segundo momento de evaluación.

- Apertura:** El docente presenta las fases del ciclo de vida que se están desarrollando, la importación de las regulación, eficiencia y efectividad de las operaciones que se realizaran junto con la implementación de los conceptos estadísticos de manera aplicada.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

• **Desarrollo:** El estudiante y docente implementan librerías dentro del software para realizar la implementación de un proceso ETL o ELT una vez realizado este proceso se implementa controles sobre los datos para la evaluación de los mismo a través de coherencia, unicidad, oportunidades y aptitud determinados a través de lenguaje de programación, con un punto de vista estadístico integrando APIs de IA para mejorar la funcionalidad y la experiencia del usuario.

• **Cierre:** El docente supervisa los equipos de trabajo y responde las preguntas o dudas que el estudiante tuviera para facilitar el aprendizaje orientado a proyectos.

Ambiente de Aprendizaje:

Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, para el desarrollo de la práctica, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> Equipo de Cómputo (PC o Lap Top) Procesadores de texto Herramientas de software digitales adecuadas para realizar las actividades solicitadas. Un entorno de desarrollo integrado (IDE) Sistema gestor de bases de datos. Framework API's Conexión a internet Video proyector Aula Virtual <p>Plataformas:</p> <ul style="list-style-type: none"> Software orientado a la unidad didáctica desarrollada, ejemplo el Classroom o Teams. <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> Libros digitales o impresos relacionados al tema. Infografías y Videos relacionados al tema. Presentaciones electrónicas relacionadas al tema. Formularios interactivos relacionados al tema. Páginas, sitios web y aplicaciones móviles relacionadas al tema. 	<p>Práctica 4: Análisis de datos.</p>	<p>Instrumento de evaluación: Rúbrica.</p> <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> Incluye una carátula con: <ul style="list-style-type: none"> Nombres de los integrantes del equipo Nombre del profesor Grupo Nombre del proyecto. Entrega de archivos electrónicos de la configuración y creación de un sitio web y aplicación móvil implementado el software requerido y su configuración en laboratorio y/o plataforma virtual. Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> Extrae, transforma y carga datos, fundamentales para preparar datos para su análisis. Implementa herramientas para cálculos estadísticos sobre los datos transformados. Aplica API's para conectar con aplicaciones de IA con el fin de realizar predicciones basadas en los datos históricos. interpreta los resultados de los análisis estadísticos para dar información para la toma de decisiones.



Unidad Didáctica 3:	Implementación de Proyectos	Nivel: Sexto
Propósito General:	Maximizar proyectos de software en sus funcionalidades ajustándolos a las necesidades del negocio a través de la mejora continua, reingeniería de software, las competencias técnicas desarrolladas en las unidades de aprendizaje del área profesional y del trabajo colaborativo, para su puesta en producción.	

Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Unidad de Competencia No 3:	Planea la entrega de un proyecto de software cumpliendo con los estándares de calidad y los requisitos técnicos y del negocio, a través de la ejecución exhaustiva de casos de prueba para validar sus funcionalidades, gestionando situaciones complejas con eficiencia en la adecuación de un sistema de información en un entorno de producción configurado que permita un rendimiento óptimo, acceso eficiente y seguro por parte de los usuarios.
Aprendizaje Esperado No 1:	Testa el proyecto de software con el fin de identificar y corregir errores antes de la puesta en producción, validando las funcionalidades requeridas y asegurando que el sistema de información cumpla con los estándares de calidad establecidos.

Tiempo estimado para obtener el Aprendizaje Esperado: 12 horas.

Contenidos de Aprendizaje

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> Técnicas de pruebas. Herramientas para la automatización de pruebas. 	<ul style="list-style-type: none"> Ejecuta pruebas manuales sobre el proyecto de software para identificar defectos y asegurar la calidad del proyecto de software. Utiliza herramientas de automatización de pruebas sobre el proyecto de software implementando scripts de prueba para garantizar la calidad del software y detectar defectos. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Responde de manera adecuada a las emociones de sus compañeros. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia Didáctica: Se usa una metodología de Aprendizaje Orientado a Proyectos (AOP).

Fase 3 de la metodología Aprendizaje Orientado a Proyectos: Implementación de tres diferentes técnicas de pruebas adecuados a la selección del proyecto.

La metodología inicia con una introducción a los conceptos fundamentales y los diferentes tipos de pruebas y herramientas para su automatización, utilizando recursos multimedia para captar el interés de los estudiantes. A continuación, se organizarán discusiones grupales sobre casos de estudio relevantes que destaquen la importancia de las pruebas en el desarrollo de software. Después, los estudiantes elaborarán un plan de pruebas para una aplicación sencilla y ejecutarán pruebas empleando herramientas.

Finalmente, se realizará una revisión donde presentarán sus resultados y reflexionarán sobre las tendencias actuales en el campo, promoviendo así un aprendizaje colaborativo y aplicable a situaciones reales.

Evaluación Formativa, se realiza una evaluación intermedia o continua para identificar las competencias que han adquirido los alumnos hasta el momento del tercer momento de evaluación.

- Apertura:** El docente explica al menos tres técnicas de pruebas viables partiendo de las necesidades del proyecto para identificar defectos y asegurar la calidad del proyecto de software.
- Desarrollo:** El docente hace una presentación electrónica y explica las diferentes técnicas de pruebas y herramientas de automatización, así como la interacción entre las mismas para trabajar de manera colaborativa posteriormente los estudiantes elaboran el plan de pruebas para ejecutarlas mediante herramientas de automatización.
- Cierre:** El docente y los estudiantes se reúnen en equipos de trabajo para verificar y retroalimentar los resultados obtenidos de las pruebas para así garantizar la calidad del software y detectar defectos.

Ambiente de Aprendizaje:



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, para el desarrollo de la práctica, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> • Equipo de Cómputo (PC o Lap Top) • Procesadores de texto • Herramientas de software digitales adecuadas para realizar las actividades solicitadas. • Un entorno de desarrollo integrado (IDE) • Conexión a internet • Video proyector • Aula Virtual • Herramientas de automatización de pruebas. <p>Plataformas:</p> <ul style="list-style-type: none"> • Software orientado a la unidad didáctica desarrollada, ejemplo el Classroom o Teams. <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Libros digitales o impresos relacionados al tema. • Infografías y Videos relacionados al tema. • Presentaciones electrónicas relacionadas al tema. • Formularios interactivos relacionados al tema. • Páginas, sitios web y aplicaciones móviles relacionadas al tema. 	<p>Práctica 5: Técnicas de pruebas.</p>	<p>Instrumento de evaluación: Rúbrica.</p> <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. • Entrega de archivos electrónicos del plan de pruebas y sus resultados en un documento pdf en laboratorio y/o plataforma virtual. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Diseña casos de prueba apropiados, exhaustivos y relevantes basados en los requerimientos del software. • Registra los resultados de las pruebas de manera clara y concisa, incluyendo los pasos realizados, los datos de entrada y los resultados obtenidos. • Da seguimiento a los defectos reportados hasta su resolución, verificando que la corrección no haya introducido nuevos problemas. • Selecciona las herramientas de automatización adecuadas para la aplicación web y móvil y los casos de prueba a automatizar. • Desarrolla scripts de prueba robustos y mantenibles, utilizando las mejores prácticas de programación. • Integra los scripts de prueba con herramientas de gestión de pruebas para facilitar la ejecución y el seguimiento de las pruebas.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Propósito General:	Maximizar proyectos de software en sus funcionalidades ajustándolos a las necesidades del negocio a través de la mejora continua, reingeniería de software, las competencias técnicas desarrolladas en las unidades de aprendizaje del área profesional y del trabajo colaborativo, para su puesta en producción.		
Unidad de Competencia No 3:	Planea la entrega de un proyecto de software cumpliendo con los estándares de calidad y los requisitos técnicos y del negocio, a través de la ejecución exhaustiva de casos de prueba para validar sus funcionalidades, gestionando situaciones complejas con eficiencia en la adecuación de un sistema de información en un entorno de producción configurado que permita un rendimiento óptimo, acceso eficiente y seguro por parte de los usuarios.		
Aprendizaje Esperado No 2:	Adapta el proyecto de software en un entorno de producción configurado correctamente para su funcionamiento óptimo conforme a los requisitos técnicos y del negocio para el acceso de los usuarios.	Tiempo estimado para obtener el Aprendizaje Esperado:	12 horas.

Contenidos de Aprendizaje

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> Fundamentos de Despliegue de Software. Arquitectura de Aplicaciones. Seguridad en el Despliegue. 	<ul style="list-style-type: none"> Configura servidores, bases de datos y recursos en la nube utilizando herramientas de automatización para agilizar el despliegue y reducir errores de configuración. Configura variables de entorno, rutas de acceso, configuraciones de red, y otros parámetros necesarios para el funcionamiento del proyecto de software en el despliegue. Usa de herramientas para automatizar la compilación y empaquetado del código. Configura los mecanismos de seguridad en el entorno de despliegue para prevenir ataques cibernéticos comunes y proteger los datos sensibles de la aplicación. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Reconoce, entiende y gestiona las propias emociones y las de los integrantes de su equipo de manera efectiva. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Responde de manera adecuada a las emociones de sus compañeros. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia Didáctica: Se usa una metodología de Aprendizaje Orientado a Proyectos (AOP).

Fase 3 y 4 de la metodología Aprendizaje Orientado a Proyectos: Implementación del proyecto de software en un entorno de producción configurando servidores y mecanismos de seguridad. Se propone un enfoque práctico que integra teoría y aplicación directa, los estudiantes adquirirán las habilidades necesarias para seleccionar y configurar tecnologías adecuadas. Se fomentará el trabajo en equipo y la resolución de problemas, creando un ambiente de aprendizaje activo donde los estudiantes podrán experimentar con la implementación de soluciones en entornos de desarrollo y producción, obteniendo así una comprensión integral de los procesos involucrados en el despliegue de software.

Evaluación Formativa, se realiza una evaluación intermedia o continua para identificar las competencias que han adquirido los alumnos hasta el momento del tercer momento de evaluación.

- Apertura:** El docente presenta los fundamentos de despliegue de software, etapas, arquitectura de aplicaciones y mecanismos de seguridad. Posteriormente ejemplifica la preparación del entorno donde el software será desplegado y de todos sus elementos necesarios: servidores, bases de datos, redes y otros componentes necesarios para el funcionamiento del proyecto de manera de que el estudiante, construya y conceptualice su propio conocimiento mediante estos ejemplos y elabore el plan de despliegue.

Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

- **Desarrollo:** El estudiante implementa las etapas de despliegue de software, configura servidores, bases de datos, servicios y recursos en la nube, utilizando herramientas de automatización, complicación y empaquetado para gestionar el proceso de despliegue y reducir los errores de configuración para elaborar el plan de despliegue del proyecto.
- **Cierre:** El docente supervisa los equipos de trabajo y responde las preguntas o dudas que el estudiante tuviera para facilitar el aprendizaje orientado a proyectos.

Ambiente de Aprendizaje:

Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, para el desarrollo de la práctica, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Instrumento y Criterios de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> • Equipo de Cómputo (PC o Lap Top) • Procesadores de texto • Herramientas de software adecuadas para realizar las actividades solicitadas. • Un entorno de desarrollo integrado (IDE) • Sistema gestor de bases de datos. • Framework • Conexión a internet • Video proyector • Aula Virtual <p>Plataformas:</p> <ul style="list-style-type: none"> • Software orientado a la unidad didáctica desarrollada, ejemplo el Classroom o Teams. <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Libros digitales o impresos relacionados al tema. • Infografías y Videos relacionados al tema. • Presentaciones electrónicas relacionadas al tema. • Formularios interactivos relacionados al tema. • Páginas, sitios web y aplicaciones móviles relacionadas al tema. 	<p>Práctica 6: Implementación de Sistema de Información.</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Rúbrica. <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. • Entrega archivo electrónico del informe de la implementación del proyecto de software en pdf en laboratorio y/o plataforma virtual. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Elabora un plan de implementación incluyendo alcance, objetivos, cronograma, recursos, riesgos, gestión de cambios, capacitación de usuarios. • Configura correctamente el entorno de producción, incluyendo servidores, bases de datos y otros componentes, y documenta la configuración. • Ejecuta procedimientos de migración de datos de manera eficiente y segura. • Ejecuta casos de prueba para verificar la funcionalidad del sistema en el entorno de producción. • Elabora manuales y tutoriales del sistema de información claros y concisos para el usuario final. • Presenta de manera clara y concisa los resultados del proyecto de software, incluyendo los logros, desafíos, contingencias y mecanismos de adaptación a cambios. • Evalúa el desempeño individual como el del equipo respecto a las prácticas involucradas en la implementación del sistema de información Superior

Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

PRÁCTICAS

Nombre de la Práctica:	Redefinición de requerimientos.	N° de la Práctica:	1	Tiempo:	6 horas.
Unidad de Competencia:	Evalúa el proyecto de software existente, a través del análisis exhaustivo de las áreas críticas de mejora y escalabilidad de procesos del negocio estableciendo el plan de desarrollo de software, empleando un pensamiento crítico para las mejoras que garanticen la eficiencia, eficacia y sostenibilidad del software con la finalidad de optimizar la toma de decisiones.				
Aprendizajes Esperados Relacionados con la Práctica:	Identifica áreas críticas de mejora, a través de un análisis exhaustivo en el proyecto de software existente para la optimización de procesos y escalabilidad de los mismos.				
Contenidos de Aprendizaje Relacionados con la Práctica					
Conceptuales	Procedimentales	Actitudinales			
<ul style="list-style-type: none"> Recopilación de requerimientos del sistema. Revisión y priorización de requerimientos. Redefinición de requerimientos. 	<ul style="list-style-type: none"> Selecciona alguna de las técnicas de priorización de requerimientos (.Método Moscow, Pirámide de Priorización, Método RICE, Modelo de Kano) al proyecto de software y redefinir los requerimientos del mismo. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido. 			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategia Didáctica: Aprendizaje Orientado a Proyectos</p> <p>Los estudiantes se reúnen para iniciar los trabajos en equipo colaborativo, retomando los requerimientos definidos de su proyecto de software realizado en el Laboratorio de Desarrollo de Software III, realizando un análisis de las funcionalidades de los procesos del negocio que cubren, los actores involucrados, dependencias y restricciones de las funcionalidades, ambigüedades e inconsistencias y empezarán a evaluar la claridad, integridad y consistencia de estos, así como los nuevos requerimientos para su escalabilidad, esto a través de la elección de técnicas de levantamiento de requerimientos con los usuarios finales.</p> <p>Posteriormente, identificarán los requerimientos esenciales del negocio y los deseables para escalar la funcionalidad del proyecto de software, a través de la aplicación de técnicas de priorización de requerimientos, como (MoSCoW, Kano, costo-beneficio) y elaborar una matriz que muestre los requerimientos, la técnica de priorización aplicada, la prioridad asignada y la justificación de dicha prioridad; con el fin de redefinir los requerimientos ambiguos e inconsistentes y los propios a los nuevos procesos del negocio.</p> <p>Los estudiantes entregan al docente la práctica de manera presencial y/o en el aula virtual.</p> <p>Ambiente de Aprendizaje:</p> <p>Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.</p>					
Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
Herramientas Tecnológicas:	<ul style="list-style-type: none"> Revisión y priorización de requerimientos. 	Instrumento de evaluación: Rúbrica			



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

<ul style="list-style-type: none"> • Equipo de Cómputo (PC o Lap Top) • Procesadores de texto • Herramientas de software digitales adecuadas para realizar las actividades solicitadas. • Un entorno de desarrollo integrado (IDE) • Conexión a internet • Video proyector • Aula Virtual <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Libros digitales o impresos relacionados al tema. • Infografías y Videos relacionados al tema. • Presentaciones electrónicas relacionadas al tema. • Formularios interactivos relacionados al tema. • Páginas, sitios web y aplicaciones móviles relacionadas al tema. 	<ul style="list-style-type: none"> • Redefinición de requerimientos. 	<p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. • Entrega de archivos electrónicos de la redefinición de requerimientos en un documento pdf en laboratorio y/o plataforma virtual. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Aplica un análisis detallado de los requerimientos actuales del proyecto, identificando las fortalezas y debilidades de los requerimientos existentes. • Demuestra un conocimiento sólido de al menos dos técnicas de priorización de requerimientos (por ejemplo, MoSCoW, Kano, costo-beneficio). • Aplica la técnica seleccionada de manera adecuada al proyecto de software, identificando correctamente los criterios de priorización. • Explica de forma clara y concisa los motivos de la priorización asignada a cada requerimiento, respaldando sus decisiones con argumentos sólidos. • Identifica correctamente los requerimientos que necesitan ser redefinidos debido a ambigüedades, inconsistencias o cambios en el contexto del proyecto de software. • Sugiere modificaciones a los requerimientos que los hagan más claros, precisos y completos, sin comprometer los objetivos del proyecto de software. • Evalúa el impacto de las modificaciones propuestas en otras partes del proyecto y en los costos y plazos.
---	---	---

Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Nombre de la Práctica:	Gestión del cambio de requerimientos.	N° de la Práctica:	2	Tiempo:	6 horas.
Unidad de Competencia:	Evalúa el proyecto de software existente, a través del análisis exhaustivo de las áreas críticas de mejora y escalabilidad de procesos del negocio estableciendo el plan de desarrollo de software, empleando un pensamiento crítico para las mejoras que garanticen la eficiencia, eficacia y sostenibilidad del software con la finalidad de optimizar la toma de decisiones.				
Aprendizajes Esperados Relacionados con la Práctica:	Estructura el plan de desarrollo de proyectos de software, incorporando cronogramas detallados y estrategias de mitigación de riesgos para implementar las mejoras y nuevos requisitos.				
Contenidos de Aprendizaje Relacionados con la Práctica					
Conceptuales	Procedimentales	Actitudinales			
<ul style="list-style-type: none"> Gestión del cambio de requerimientos. Planificación de las tareas del desarrollo de software. Gráfica de Gantt y ruta crítica. 	<ul style="list-style-type: none"> Establece la gestión de cambios de requerimientos del proyecto de software existente. Reestructura el diagrama de Gantt y la ruta crítica del proyecto de software existente. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido. 			
Estrategia Didáctica y Ambiente de Aprendizaje					
<p>Estrategia Didáctica: Aprendizaje Orientado a Proyectos</p> <p>Los estudiantes se reúnen para continuar con los trabajos en equipo colaborativo, para tomar la redefinición de los requerimientos del proyecto de software con el fin de realizar la planeación de este, contemplando las fases del ciclo de vida del software, las tareas a realizar de cada fase, recursos, costos y fechas de inicio y culminación de cada tarea y generar la gráfica de Gantt que permita el seguimiento de cada tarea y su culminación.</p> <p>Elaborarán la ruta crítica con la interrelación de las tareas del plan del proyecto de software para identificar las tareas más importantes del proyecto.</p> <p>Los estudiantes entregan al docente la práctica de manera presencial y/o en el aula virtual.</p> <p>Ambiente de Aprendizaje:</p> <p>Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.</p>					
Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación			
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> Equipo de Cómputo (PC o Lap Top) Procesadores de texto Herramientas de software digitales adecuadas para realizar las actividades solicitadas. Un entorno de desarrollo integrado (IDE) 	<p>Diagrama de Gantt y ruta crítica del proyecto de software existente.</p>	<p>Instrumento de evaluación: Rúbrica</p> <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> Incluye una carátula con: 			





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

<ul style="list-style-type: none"> • Conexión a internet • Video proyector • Aula Virtual <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Libros digitales o impresos relacionados al tema. • Infografías y Vídeos relacionados al tema. • Presentaciones electrónicas relacionadas al tema. • Formularios interactivos relacionados al tema. • Páginas, sitios web y aplicaciones móviles relacionadas al tema. 		<ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. <ul style="list-style-type: none"> • Entrega de archivos electrónicos del documento del plan de Gestión del cambio de requerimientos en un documento pdf en laboratorio y/o plataforma virtual. • Entrega de archivos electrónicos Gráfica de Gantt y ruta crítica en un documento pdf en laboratorio y/o plataforma virtual. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Demuestra un entendimiento profundo de los procesos y procedimientos involucrados en la gestión de cambios de requerimientos, incluyendo la identificación, evaluación, aprobación y seguimiento de los mismos. • Desarrolla un plan de gestión de cambios claro y conciso, que incluya roles y responsabilidades, procedimientos de solicitud y aprobación, y mecanismos de seguimiento. • Construye una matriz de rastreabilidad que permita vincular los requerimientos con otros elementos del proyecto de software y facilitar el seguimiento de los cambios. • Analiza el impacto de los cambios de requerimientos en el cronograma del proyecto de software e identifica las tareas afectadas. • Actualiza el diagrama de Gantt y la ruta crítica de manera precisa para reflejar los cambios realizados. • Propone ajustes en la asignación de recursos para minimizar el impacto de los cambios en el cronograma.
---	--	--



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Nombre de la Práctica:	Ciclo de vida de los datos.	N° de la Práctica:	3	Tiempo:	12 horas.
Unidad de Competencia:	Mejora el proyecto de software a través de una reestructuración de la arquitectura, la refactorización del código y la integración de tecnologías emergentes, con el fin de optimizar el rendimiento, escalabilidad, mantenibilidad y capacidad de adaptación, facilitando así la toma de decisiones basadas en análisis estadísticos y proyecciones basadas en la implementación de la inteligencia artificial				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea el lenguaje de modelado unificado (UML), en el rediseño de interacciones de los usuarios con el sistema, la arquitectura del software y el flujo de los procesos para mejorar su eficiencia, escalabilidad y sostenibilidad.				

Contenidos de Aprendizaje Relacionados con la Práctica

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> Lenguaje de modelado unificado para el diseño del software. Herramientas de diagramación UML. Modelado de la base de datos relacional o no relacional del proyecto de software. Normalización de bases de datos relacionales. 	<ul style="list-style-type: none"> Modela el diseño, arquitectura, flujos, componentes y despliegue del proyecto de software web conforme al estándar UML para garantizar una representación precisa y comprensible del sistema. Diseña interfaces gráficas web intuitivas y visualmente atractivas para optimizar la experiencia del usuario. Diseña la interfaz gráfica de la aplicación móvil que facilite la interacción del usuario con las funcionalidades del sistema en dispositivos portátiles. Diseña bases de datos relacionales utilizando las formas normales con el objetivo de minimizar la redundancia y maximizar la consistencia de los datos del proyecto. Diseña bases de datos no relacionales que se adapten las necesidades cambiantes de los datos para garantizar la escalabilidad y la eficiencia del sistema. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Desarrolla soluciones creativas. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Responde de manera adecuada a las emociones de sus compañeros. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia Didáctica: Aprendizaje Orientado a Proyectos

Los estudiantes se reúnen para continuar con los trabajos en equipo colaborativo, tomando la redefinición de requerimientos para elaborar los diagramas de casos de uso para obtener las interacciones de los usuarios con el sistema, los diagramas de actividades para modelar los flujos de trabajo del sistema, los diagramas de clases para representar la estructura del sistema, los diagramas de secuencia para ver interacciones entre objetos para representar los flujos de trabajo del sistema, los Diagramas de Componentes para distinguir los componentes físicos del sistema y sus relaciones, los diagramas de despliegue para epresentar la arquitectura física del sistema, incluyendo los nodos, hardware y conexione; estableciendo las relaciones entre los diferentes diagramas UML para garantizar la coherencia del modelo.

También realizan la arquitectura del sistema que optimice el rendimiento, la escalabilidad y la mantenibilidad, considerando patrones de diseño y mejores prácticas para la arquitectura de software.

Y diseñan una interfaz gráfica intuitiva y con experiencia de usuario, tanto para la versión web como para la móvil, utilizando herramientas de diseño de interfaces para crear prototipos interactivos.

Los estudiantes entregan al docente la práctica de manera presencial y/o en el aula virtual.

Ambiente de Aprendizaje:

Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> Equipo de Cómputo (PC o Lap Top) Procesadores de texto Herramientas de software digitales adecuadas para realizar las actividades solicitadas. Un entorno de desarrollo integrado (IDE) Software de UML Conexión a internet Video proyector Aula Virtual <p>Plataformas:</p> <ul style="list-style-type: none"> Software orientado a la unidad didáctica desarrollada, ejemplo el Classroom o Teams. <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> Libros digitales o impresos relacionados al tema. Infografías y Vídeos relacionados al tema. Presentaciones electrónicas relacionadas al tema. Formularios interactivos relacionados al tema. <p>Páginas, sitios web y aplicaciones móviles relacionadas al tema.</p>	<ul style="list-style-type: none"> Modelado de arquitectura, flujos, componentes, base de datos y diseño de la interfaz gráfica de la aplicación web y móvil. Modelo de reestructuración del proyecto de software. 	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> Rúbrica. <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> Incluye una carátula con: <ul style="list-style-type: none"> Nombres de los integrantes del equipo Nombre del profesor Grupo Nombre del proyecto. Entrega de archivos electrónicos modelo de reestructuración del proyecto. (usa alguna herramienta de modelado), en un documento pdf en laboratorio y/o plataforma virtual. <p>Entrega de actividades en tiempo y forma solicitados.</p> <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> Modela la documentación de forma clara, concisa y precisa, utilizando el lenguaje de modelado unificado UML, y evitando ambigüedades. Refleja el modelo de manera precisa los requerimientos funcionales y no funcionales del sistema, demostrando una clara comprensión de estos. Propone modelos arquitectónicos y de diseño que optimicen el rendimiento, la escalabilidad y la mantenibilidad del sistema web y móvil. Diseña la interfaz gráfica intuitiva y con una experiencia de usuario, tanto para la versión web como para la móvil, considerando las características específicas de cada sistema operativo. Interrelaciona de forma lógica y coherente la arquitectura, flujos, componentes, base de datos e interfaz gráfica.

Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Nombre de la Práctica:	Análisis de datos.	N° de la Práctica:	4	Tiempo:	12 horas.
Unidad de Competencia:	Mejora el proyecto de software a través de una reestructuración de la arquitectura, la refactorización del código y la integración de tecnologías emergentes, con el fin de optimizar el rendimiento, escalabilidad, mantenibilidad y capacidad de adaptación, facilitando así la toma de decisiones basadas en análisis estadísticos y proyecciones con la implementación de la inteligencia artificial				
Aprendizajes Esperados Relacionados con la Práctica:	Establece nuevas funcionalidades y tecnologías, en el proyecto de software, a través de una refactorización del código, para mejorar su rendimiento, funcionalidad, calidad, modularidad, seguridad y mantenibilidad				

Contenidos de Aprendizaje Relacionados con la Práctica

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> Lenguaje de programación. Lenguaje de programación móvil. Entorno de Desarrollo Integrado. Framework. API's. 	<ul style="list-style-type: none"> Optimiza la codificación del proyecto mediante el uso de un IDE para aumentar la productividad y la calidad del código. Emplea frameworks para acelerar el desarrollo y garantizar la escalabilidad de proyectos. Reescribe el código para el proyecto de software mediante un lenguaje de programación web para la solución a la problemática planteada. Emplea técnicas estadísticas y de modelado predictivo en los datos para tomar decisiones y realizar pronósticos. Integra APIs de IA de manera eficiente para mejorar la funcionalidad y la experiencia del usuario. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Desarrolla soluciones creativas. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Responde de manera adecuada a las emociones de sus compañeros. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia Didáctica: Aprendizaje Orientado a Proyectos

Los estudiantes se reúnen para continuar con los trabajos en equipo colaborativo, desarrollarán el proceso utilizando un lenguaje de programación, bases de datos relacional o no relacional rediseño web, desarrollo móvil e integración de APIs de IA. Comenzarán extrayendo datos de diversas fuentes, como archivos CSV, JSON y APIs, para luego transformarlos mediante técnicas de limpieza y agregación que mejoren la calidad de los datos. Posteriormente, cargarán la información procesada en una base de datos, donde realizarán un análisis estadístico utilizando bibliotecas. A través de este análisis, identificarán patrones y tendencias en los datos, visualizando los resultados con herramientas de gráficos. Finalmente, los estudiantes reflexionarán sobre la importancia del proceso ETL en la toma de decisiones basada en datos, evaluando cómo cada etapa impacta la calidad del análisis realizado.

Los estudiantes entregan al docente la práctica de manera presencial y/o en el aula virtual.

Ambiente de Aprendizaje:

Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> • Equipo de Cómputo (PC o Lap Top) • Procesadores de texto • Herramientas de software digitales adecuadas para realizar las actividades solicitadas. • Un entorno de desarrollo integrado (IDE) • Framework • API's • Sistema gestor de bases de datos • Conexión a internet • Video proyector • Aula Virtual <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Libros digitales o impresos relacionados al tema. • Infografías y Vídeos relacionados al tema. • Presentaciones electrónicas relacionadas al tema. • Formularios interactivos relacionados al tema. • Páginas, sitios web y aplicaciones móviles relacionadas al tema. 	<ul style="list-style-type: none"> • Proceso ETL o ELT. 	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Rúbrica. <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. • Entrega de archivos electrónicos de la configuración y creación de un sitio web y aplicación móvil implementado el software requerido y su configuración en laboratorio y/o plataforma virtual. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Extrae, transforma y carga datos, fundamentales para preparar datos para su análisis. • Implementa herramientas para cálculos estadísticos sobre los datos transformados. • Aplica API's para conectar con aplicaciones de IA con el fin de realizar predicciones basadas en los datos históricos. • Interpreta los resultados de los análisis estadísticos para dar información para la toma de decisiones.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Nombre de la Práctica:	Técnicas de pruebas.	N° de la Práctica:	5	Tiempo:	8 horas.
Unidad de Competencia:	Planea la entrega de un proyecto de software cumpliendo con los estándares de calidad y los requisitos técnicos y del negocio, a través de la ejecución exhaustiva de casos de prueba para validar sus funcionalidades, gestionando situaciones complejas con eficiencia en la adecuación de un sistema de información en un entorno de producción configurado que permita un rendimiento óptimo, acceso eficiente y seguro por parte de los usuarios.				
Aprendizajes Esperados Relacionados con la Práctica:	Emplea técnicas de pruebas para su proyecto de software con el fin de identificar y corregir errores antes de la puesta en producción, validando las funcionalidades requeridas y asegurando que el sistema de información cumpla con los estándares de calidad establecidos.				

Contenidos de Aprendizaje Relacionados con la Práctica

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> Técnicas de pruebas. Herramientas para la automatización de pruebas. 	<ul style="list-style-type: none"> Ejecuta pruebas manuales sobre el proyecto de software para identificar defectos y asegurar la calidad del proyecto de software. Utiliza herramientas de automatización de pruebas sobre el proyecto de software implementando scripts de prueba para garantizar la calidad del software y detectar defectos. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Responde de manera adecuada a las emociones de sus compañeros. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia Didáctica: Aprendizaje Orientado a Proyectos

Los estudiantes se reúnen para continuar con los trabajos en equipo colaborativo, para diseñar casos de prueba para verificar que el software realice las funciones especificadas en los requerimientos, que evalúen los aspectos del rendimiento, la seguridad y su usabilidad, que verifiquen la interacción entre los diferentes componentes del sistema y evaluar el funcionamiento del sistema completo que cumplan con las expectativas del cliente. Priorizarán los casos de prueba en función de su criticidad y del riesgo asociado a cada uno.

Aplican herramientas de automatización adecuadas para la aplicación web y móvil, considerando factores como el lenguaje de programación, el framework de pruebas y la complejidad de los casos de prueba con scripts elaborados por ellos, para las pruebas utilizando las mejores prácticas de programación, asegurando su legibilidad, mantenibilidad y reusabilidad; registrando los resultados obtenidos para reportar los defectos encontrados y darle seguimiento a su corrección.

Los estudiantes entregan al docente la práctica de manera presencial y/o en el aula virtual.

Ambiente de Aprendizaje:

Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir sus avances de los proyectos de software.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> • Equipo de Cómputo (PC o Lap Top) • Procesadores de texto • Herramientas para técnicas de pruebas. • Archivo electrónico con el plan de pruebas • Un entorno de desarrollo integrado (IDE) • Software de UML • Conexión a internet • Video proyector • Aula Virtual <p>Plataformas:</p> <ul style="list-style-type: none"> • Software orientado a la unidad didáctica desarrollada, ejemplo el Classroom o Teams. <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Libros digitales o impresos relacionados al tema. • Infografías y Videos relacionados al tema. • Presentaciones electrónicas relacionadas al tema. • Formularios interactivos relacionados al tema. <p>Páginas, sitios web y aplicaciones móviles relacionadas al tema.</p>	<ul style="list-style-type: none"> • Plan de pruebas e implementación de las técnicas de prueba. • Análisis de casos, informes de pruebas realizadas y conclusiones sobre el proceso de prueba. 	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Rúbrica. <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. • Entrega de archivos electrónicos del plan de pruebas y sus resultados en un documento pdf en laboratorio y/o plataforma virtual. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Diseña casos de prueba apropiados, exhaustivos y relevantes basados en los requerimientos del software. • Registra los resultados de las pruebas de manera clara y concisa, incluyendo los pasos realizados, los datos de entrada y los resultados obtenidos. • Da seguimiento a los defectos reportados hasta su resolución, verificando que la corrección no haya introducido nuevos problemas. • Selecciona las herramientas de automatización adecuadas para la aplicación web y móvil y los casos de prueba a automatizar. • Desarrolla scripts de prueba robustos y mantenibles, utilizando las mejores prácticas de programación. • Integra los scripts de prueba con herramientas de gestión de pruebas para facilitar la ejecución y el seguimiento de las pruebas.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Nombre de la Práctica:	Implementación de Sistema de Información.	N° de la Práctica:	6	Tiempo:	8 horas.
Unidad de Competencia:	Planea la entrega de un proyecto de software cumpliendo con los estándares de calidad y los requisitos técnicos y del negocio, a través de la ejecución exhaustiva de casos de prueba para validar sus funcionalidades, gestionando situaciones complejas con eficiencia en la adecuación de un sistema de información en un entorno de producción configurado que permita un rendimiento óptimo, acceso eficiente y seguro por parte de los usuarios.				
Aprendizajes Esperados Relacionados con la Práctica:	Elabora el plan de implementación del proyecto de software en un entorno de producción configurado correctamente para su funcionamiento óptimo conforme a los requisitos técnicos y del negocio para el acceso de los usuarios.				

Contenidos de Aprendizaje Relacionados con la Práctica

Conceptuales	Procedimentales	Actitudinales
<ul style="list-style-type: none"> Fundamentos de Despliegue de Software. Etapas de plan de despliegue de software. Arquitectura de Aplicaciones. <ol style="list-style-type: none"> Web Mobil Patrones de arquitectura. Seguridad en el Despliegue. Mecanismos de seguridad 	<ul style="list-style-type: none"> Configura servidores, bases de datos y recursos en la nube utilizando herramientas de automatización para agilizar el despliegue y reducir errores de configuración. Configura variables de entorno, rutas de acceso, configuraciones de red, y otros parámetros necesarios para el funcionamiento del proyecto de software en el despliegue. Usa de herramientas para automatizar la compilación y empaquetado del código. Configura los mecanismos de seguridad en el entorno de despliegue para prevenir ataques cibernéticos comunes y proteger los datos sensibles de la aplicación. 	<ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Reconoce, entiende y gestiona las propias emociones y las de los integrantes de su equipo de manera efectiva. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Responde de manera adecuada a las emociones de sus compañeros. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido.

Estrategia Didáctica y Ambiente de Aprendizaje

Estrategia Didáctica: Aprendizaje Orientado a Proyectos

Los estudiantes se reúnen para continuar con los trabajos en equipo colaborativo con el objetivo de elegir un sistema de hosteo adecuado a las características del sistema de información y los requisitos del cliente, para pasar a realizar la configuración del servidor, el lenguaje de programación, el gestor de bases de datos, dominio del servicio, certificado de seguridad, entre otros, en el servicio de hosting.

Configurarán las variables de entorno, las rutas de acceso, las configuraciones de red y otros parámetros necesarios para el funcionamiento del sistema de información en la nube, y pasan a realizar las pruebas exhaustivas para verificar el correcto funcionamiento de la aplicación en el entorno de producción.

Desarrollan materiales de capacitación, planificación de sesiones de entrenamiento para el usuario final con respecto al uso del sistema de información, y elaboran un informe del plan de implementación y hacer la presentación del mismo.

Los estudiantes entregan al docente la práctica de manera presencial y/o en el aula virtual.

Ambiente de Aprendizaje:

Es un ambiente de aprendizaje presencial en los laboratorios del área de programación, adicionalmente el docente creará un aula virtual en plataformas digitales con el estudiante con la finalidad de dejar materiales didácticos para la consulta de los estudiantes fuera de los tiempos establecidos de la clase y poder recibir los avances de los proyectos de software.



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Herramientas Tecnológicas y Recursos Didácticos	Evidencia de Aprendizaje para la Evaluación Formativa	Criterios e Instrumentos de Evaluación
<p>Herramientas Tecnológicas:</p> <ul style="list-style-type: none"> • Equipo de Cómputo (PC o Lap Top) • Procesadores de texto • Herramientas para técnicas de pruebas. • Archivo electrónico con el plan de pruebas • Un entorno de desarrollo integrado (IDE) • Software de UML • Conexión a internet • Video proyector • Aula Virtual <p>Plataformas:</p> <ul style="list-style-type: none"> • Software orientado a la unidad didáctica desarrollada, ejemplo el Classroom o Teams. <p>Recursos Didácticos:</p> <ul style="list-style-type: none"> • Libros digitales o impresos relacionados al tema. • Infografías y Videos relacionados al tema. • Presentaciones electrónicas relacionadas al tema. • Formularios interactivos relacionados al tema. <p>Páginas, sitios web y aplicaciones móviles relacionadas al tema.</p>	<ul style="list-style-type: none"> • Implementación del proyecto de Software. 	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Rúbrica. <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. • Entrega archivo electrónico del informe de la implementación del proyecto de software en pdf en laboratorio y/o plataforma virtual. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Elabora un plan de implementación incluyendo alcance, objetivos, cronograma, recursos, riesgos, gestión de cambios, capacitación de usuarios. • Configura correctamente el entorno de producción, incluyendo servidores, bases de datos y otros componentes, y documenta la configuración. • Ejecuta procedimientos de migración de datos de manera eficiente y segura. • Ejecuta casos de prueba para verificar la funcionalidad del sistema en el entorno de producción. • Elabora manuales y tutoriales del sistema de información claros y concisos para el usuario final. • Presenta de manera clara y concisa los resultados del proyecto de software, incluyendo los logros, desafíos, contingencias y mecanismos de adaptación a cambios. • Evalúa el desempeño individual como el del equipo respecto a las prácticas involucradas en la implementación del sistema de información.



PLAN DE EVALUACIÓN SUMATIVA

N°	Unidad de Competencia	Evidencia Integradora	Criterios e Instrumento de Evaluación	Porcentaje de Acreditación
1	<p>Evalúa el proyecto de software existente, a través del análisis exhaustivo de las áreas críticas de mejora y escalabilidad de procesos del negocio estableciendo el plan de desarrollo de software, empleando un pensamiento crítico para las mejoras que garanticen la eficiencia, eficacia y sostenibilidad del software con la finalidad de optimizar la toma de decisiones.</p>	<p>Rediseño y gestión integral de un plan de proyecto de software existente.</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Rúbrica. <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. • Entrega de archivos electrónicos en formato pdf en laboratorio y/o plataforma virtual, de: <ul style="list-style-type: none"> ○ Redefinición de requerimientos. ○ plan de Gestión del cambio de requerimientos ○ Gráfica de Gantt y Ruta crítica • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Revisa y prioriza los requerimientos del proyecto de software de manera efectiva, considerando su importancia, complejidad y viabilidad. • Adapta los requerimientos iniciales a las nuevas necesidades del negocio. • Planifica y gestiona el proyecto de software mediante la creación de un diagrama de Gantt y la identificación de la ruta crítica, asegurando la optimización de los recursos y el cumplimiento de los plazos. 	<p>20%</p>
2	<p>Mejora el proyecto de software a través de una reestructuración de la arquitectura, la refactorización del código y la integración de tecnologías emergentes, con el fin de optimizar el rendimiento, escalabilidad, mantenibilidad y capacidad de adaptación, facilitando así la toma de decisiones basadas en análisis estadísticos y proyecciones con la implementación de la inteligencia artificial.</p>	<p>Diseño y desarrollo integral de una solución de software.</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Rúbrica. <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo 	<p>50%</p>  <p>INSTITUTO POLITÉCNICO NACIONAL Dirección de Educación Media Superior</p>



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

			<ul style="list-style-type: none"> ○ Nombre del proyecto. • Entrega de archivos electrónicos en formato pdf en laboratorio y/o plataforma virtual, de: <ul style="list-style-type: none"> ○ modelo de reestructuración del proyecto. ○ configuración y creación de un sitio web y aplicación móvil. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Diseña y crea una solución de software integral, incluyendo la arquitectura, los flujos de trabajo, los componentes, la base de datos y las interfaces gráficas para aplicaciones web y móviles asegurando la coherencia y la funcionalidad del sistema. • Elabora un modelo de reestructuración que optimice la arquitectura, el código y la base de datos del proyecto de software existente. • Implementa un proceso ETL o ELT eficiente, utilizando las bibliotecas adecuadas para extraer, transformar y cargar datos, y realizar análisis estadísticos pertinentes. 	
3	<p>Planea la entrega de un proyecto de software cumpliendo con los estándares de calidad y los requisitos técnicos y del negocio, a través de la ejecución exhaustiva de casos de prueba para validar sus funcionalidades, gestionando situaciones complejas con eficiencia en la adecuación de un sistema de información en un entorno de producción configurado que permita un rendimiento óptimo, acceso eficiente y seguro por parte de los usuarios.</p>	<p>Verificación y despliegue del proyecto de software.</p>	<p>Instrumento de evaluación:</p> <ul style="list-style-type: none"> • Rúbrica. <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> • Incluye una carátula con: <ul style="list-style-type: none"> ○ Nombres de los integrantes del equipo ○ Nombre del profesor ○ Grupo ○ Nombre del proyecto. • Entrega de archivos electrónicos en formato pdf en laboratorio y/o plataforma virtual, de: <ul style="list-style-type: none"> ○ plan de pruebas y sus resultados. ○ informe de la implementación del proyecto de software. • Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> • Verifica y valida el funcionamiento del software mediante la creación y ejecución de 	<p style="text-align: right;">30%</p>  <p style="text-align: right;">IPN INSTITUTO POLITÉCNICO NACIONAL Dirección de</p>

Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Propósito de la Unidad de Aprendizaje	Evidencia Integradora	Criterios e Instrumento de Evaluación	Porcentaje de Acreditación
<p>Maximiza proyectos de software en sus funcionalidades ajustándolos a las necesidades del negocio a través de la mejora continua, reingeniería de software, las competencias técnicas desarrolladas en las unidades de aprendizaje del área profesional y del trabajo colaborativo, para su puesta en producción.</p>	<p>Despliegue del sistema de información en un entorno real.</p>	<p>Instrumento de Evaluación:</p> <ul style="list-style-type: none"> Rúbrica <p>Criterios de Forma (estilo):</p> <ul style="list-style-type: none"> Entrega de archivos electrónicos en formato pdf en laboratorio y/o plataforma virtual, de: <ul style="list-style-type: none"> Redefinición de requerimientos. plan de Gestión del cambio de requerimientos Gráfica de Gantt y Ruta crítica modelo de reestructuración del proyecto. configuración y creación de un sitio web y aplicación móvil. plan de pruebas y sus resultados. informe de la implementación del proyecto de software. Entrega de actividades en tiempo y forma solicitados. <p>Criterios de Fondo (parte técnica):</p> <ul style="list-style-type: none"> Revisa y prioriza los requerimientos del proyecto de software de manera efectiva. 	<p>100%</p>  <p>INSTITUTO POLITÉCNICO NACIONAL Dirección de Educación Media Superior</p>



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

		<p>considerando su importancia, complejidad y viabilidad.</p> <ul style="list-style-type: none"> • Adapta los requerimientos iniciales a las nuevas necesidades del negocio. • Planifica y gestiona el proyecto de software mediante la creación de un diagrama de Gantt y la identificación de la ruta crítica, asegurando la optimización de los recursos y el cumplimiento de los plazos. • Diseña y crea una solución de software integral, incluyendo la arquitectura, los flujos de trabajo, los componentes, la base de datos y las interfaces gráficas para aplicaciones web y móviles asegurando la coherencia y la funcionalidad del sistema. • Elabora un modelo de reestructuración que optimice la arquitectura, el código y la base de datos del proyecto de software existente. • Implementa un proceso ETL o ELT eficiente, utilizando las bibliotecas adecuadas para extraer, transformar y cargar datos, y realizar análisis estadísticos pertinentes. • Verifica y valida el funcionamiento del software mediante la creación y ejecución de un plan de pruebas exhaustivo, aplicando herramientas de automatización de pruebas. • Evalúa la calidad del software a través del análisis de casos de prueba y propone mejoras basadas en los resultados obtenidos. • Pone en marcha el proyecto de software en un entorno operativo, asegurando su correcto funcionamiento. • Documenta el sistema de información de manera clara y concisa, creando manuales y tutoriales adaptados a las necesidades del usuario final. • Presenta de manera clara y concisa los resultados del proyecto, incluyendo logros, desafíos, contingencias y mecanismos de adaptación a cambios. 	 <p>INSTITUTO POLITÉCNICO NACIONAL Dirección de Educación Media Superior</p>
--	--	--	--

PROGRAMA SINTÉTICO

PROPÓSITO DE LA UNIDAD DE APRENDIZAJE

Maximiza proyectos de software en sus funcionalidades ajustándolos a las necesidades del negocio a través de la mejora continua, reingeniería de software, las competencias técnicas desarrolladas en las unidades de aprendizaje del área profesional y del trabajo colaborativo, para su puesta en producción.

N°	UNIDAD DE COMPETENCIA	APRENDIZAJES ESPERADOS	CONTENIDOS DE APRENDIZAJE/SABERES
1	<p>Evalúa el proyecto de software existente, a través del análisis exhaustivo de las áreas críticas de mejora y escalabilidad de procesos del negocio estableciendo el plan de desarrollo de software, empleando un pensamiento crítico para las mejoras que garanticen la eficiencia, eficacia y sostenibilidad del software con la finalidad de optimizar la toma de decisiones.</p>	<p>1. Identifica áreas críticas de mejora, a través de un análisis exhaustivo en el proyecto de software existente para la optimización de procesos y escalabilidad de los mismos.</p> <p>2. Estructura el plan de desarrollo de proyectos de software, incorporando cronogramas detallados y estrategias de mitigación de riesgos para implementar las mejoras y nuevos requisitos.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> Recopilación de requerimientos del sistema. Evaluación de eficiencia y eficacia de los requerimientos. Revisión y priorización de requerimientos. Redefinición de requerimientos. <p>Procedimental:</p> <ul style="list-style-type: none"> Aplica técnicas de levantamiento de requerimientos de software con los usuarios finales para asegurar que el software cumpla con sus necesidades. Redacta documentos de requisitos detallados y estructurados para asegurar la comprensión y ejecución exitosa de un proyecto de software. Realiza pruebas de aceptación rigurosas para para garantizar la conformidad de los nuevos requisitos. <p>Actitudinal:</p> <ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido. <p>Conceptual:</p> <ul style="list-style-type: none"> Gestión del cambio de requerimientos. Planificación de las tareas del desarrollo de software. Gráfica de Gantt y ruta crítica.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

			<p>Procedimental:</p> <ul style="list-style-type: none"> • Elabora la planificación de las tareas del desarrollo del software para responder a los cambios y garantizar la calidad del sistema. • Evalúa lenguajes de programación, frameworks, API's, hardware, software, redes y plataformas utilizadas apropiados para optimizar el rendimiento y la escalabilidad del proyecto del software. <p>Actitudinal:</p> <ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Analiza información de manera crítica. • Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad. • Se comunica asertivamente y respetuosa. • Responde de manera adecuada a las emociones de sus compañeros. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido.
2	<p>Mejora el proyecto de software a través de una reestructuración de la arquitectura, la refactorización del código y la integración de tecnologías emergentes, con el fin de optimizar el rendimiento, escalabilidad, mantenibilidad y capacidad de adaptación, facilitando así la toma de decisiones basadas en análisis estadísticos y proyecciones con la implementación de la inteligencia artificial.</p>	<p>1. Emplea el lenguaje de modelado unificado (UML), en el rediseño de interacciones de los usuarios con el sistema, la arquitectura del software y el flujo de los procesos para mejorar su eficiencia, escalabilidad y sostenibilidad.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> • Metodología de desarrollo ágil. • Patrones de Diseño. • Arquitecturas de software. • Lenguaje de modelado unificado para el diseño del software. • Herramientas de diagramación UML. • Modelado de la base de datos relacional o no relacional del proyecto de software. • Normalización de bases de datos relacionales. <p>Procedimental:</p> <ul style="list-style-type: none"> • Modela el diseño, arquitectura, flujos, componentes y despliegue del proyecto de software web conforme al estándar UML para garantizar una representación precisa y comprensible del sistema. • Diseña interfaces gráficas web intuitivas y visualmente atractivas para optimizar la experiencia del usuario.





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

			<ul style="list-style-type: none"> • Diseña la interfaz gráfica de la aplicación móvil que facilite la interacción del usuario con las funcionalidades del sistema en dispositivos portátiles. • Diseña bases de datos relacionales utilizando las formas normales con el objetivo de minimizar la redundancia y maximizar la consistencia de los datos del proyecto. • Diseña bases de datos no relacionales que se adapten las necesidades cambiantes de los datos para garantizar la escalabilidad y la eficiencia del sistema. <p>Actitudinal:</p> <ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Desarrolla soluciones creativas. • Analiza información de manera crítica. • Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad. • Se comunica asertivamente y respetuosa. • Responde de manera adecuada a las emociones de sus compañeros. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido.
		<p>2. Establece nuevas funcionalidades y tecnologías, en el proyecto de software, a través de una refactorización del código, para mejorar su rendimiento, funcionalidad, calidad, modularidad, seguridad y mantenibilidad.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> • Lenguaje de programación. • Lenguaje de programación móvil. • Entorno de Desarrollo Integrado. • Frameworks. • Entornos de ejecución del lado del servidor. • Conceptos estadísticos, distribuciones, regresiones, y análisis descriptivo. • Modelos Predictivos. • Criptografía y autenticación. • Fundamentos de IA. <p>Procedimental:</p> <ul style="list-style-type: none"> • Optimiza la codificación del proyecto mediante el uso de un IDE para aumentar la productividad y la calidad del código.  <p>INSTITUTO POLITÉCNICO NACIONAL Dirección de Educación Media Superior</p>



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

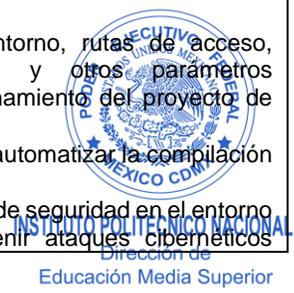
			<ul style="list-style-type: none"> • Emplea frameworks para acelerar el desarrollo y garantizar la escalabilidad de proyectos. • Reescribe el código para el proyecto de software mediante un lenguaje de programación web para la solución a la problemática planteada. • Emplea un lenguaje de programación móvil específico, adaptado a las necesidades del proyecto, para desarrollar una aplicación móvil funcional y eficiente. • Conecta una aplicación a una base de datos (relacional o no relacional) con el fin de almacenar y recuperar datos. • Emplea técnicas estadísticas y de modelado predictivo en los datos para tomar decisiones y realizar pronósticos. • Aplica técnicas de encriptación, autenticación, y control de acceso en el software con el fin de garantizar la seguridad del software. • Integra APIs de IA de manera eficiente para para mejorar la funcionalidad y la experiencia del usuario. <p>Actitudinal:</p> <ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Desarrolla soluciones creativas. • Analiza información de manera crítica. • Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad. • Se comunica asertivamente y respetuosa. • Responde de manera adecuada a las emociones de sus compañeros. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido.
3	<p>Planea la entrega de un proyecto de software cumpliendo con los estándares de calidad y los requisitos técnicos y del negocio, a través de la ejecución exhaustiva de casos de prueba para validar sus funcionalidades, gestionando situaciones complejas con eficiencia en la adecuación de un sistema de información en un entorno de producción configurado que permita un</p>	<p>1. Testa el proyecto de software con el fin de identificar y corregir errores antes de la puesta en producción, validando las funcionalidades requeridas y asegurando que el sistema de información cumpla con los estándares de calidad establecidos.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> • Técnicas de pruebas. • Herramientas para la automatización de pruebas.  <p>INSTITUTO POLITÉCNICO NACIONAL Dirección de Educación Media Superior</p>



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

	<p>rendimiento óptimo, acceso eficiente y seguro por parte de los usuarios.</p>		<p>Procedimental:</p> <ul style="list-style-type: none"> Ejecuta pruebas manuales sobre el proyecto de software para identificar defectos y asegurar la calidad del proyecto de software. Utiliza herramientas de automatización de pruebas sobre el proyecto de software implementando scripts de prueba para garantizar la calidad del software y detectar defectos. <p>Actitudinal:</p> <ul style="list-style-type: none"> Coopera activamente en equipos de trabajo. Analiza información de manera crítica. Resuelve problemas de forma efectiva. Guía a su equipo en el logro de objetivos. Se adapta a cambios y desafíos. Demuestra sentido de la responsabilidad. Se comunica asertivamente y respetuosa. Responde de manera adecuada a las emociones de sus compañeros. Utiliza herramientas tecnológicas para resolver problemas. Muestra un comportamiento ético y comprometido.
		<p>2. Adapta el proyecto de software en un entorno de producción configurado correctamente para su funcionamiento óptimo conforme a los requisitos técnicos y del negocio para el acceso de los usuarios.</p>	<p>Conceptual:</p> <ul style="list-style-type: none"> Fundamentos de Despliegue de Software. Arquitectura de Aplicaciones. Seguridad en el Despliegue. <p>Procedimental:</p> <ul style="list-style-type: none"> Configura servidores, bases de datos y recursos en la nube utilizando herramientas de automatización para agilizar el despliegue y reducir errores de configuración. Configura variables de entorno, rutas de acceso, configuraciones de red, y otros parámetros necesarios para el funcionamiento del proyecto de software en el despliegue. Usa de herramientas para automatizar la compilación y empaquetado del código. Configura los mecanismos de seguridad en el entorno de despliegue para prevenir ataques cibernéticos





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

			<p>comunes y proteger los datos sensibles de la aplicación.</p> <p>Actitudinal:</p> <ul style="list-style-type: none"> • Coopera activamente en equipos de trabajo. • Reconoce, entiende y gestiona las propias emociones y las de los integrantes de su equipo de manera efectiva. • Analiza información de manera crítica. Resuelve problemas de forma efectiva. • Guía a su equipo en el logro de objetivos. • Se adapta a cambios y desafíos. • Demuestra sentido de la responsabilidad. • Se comunica asertivamente y respetuosa. • Responde de manera adecuada a las emociones de sus compañeros. • Utiliza herramientas tecnológicas para resolver problemas. • Muestra un comportamiento ético y comprometido.
--	--	--	---



BIBLIOGRAFÍA BÁSICA Y COMPLEMENTARIA

Número y Nombre de la Unidad Didáctica	FORMATO APA	CLASIFICACIÓN	
		Básico	Consulta
Unidad 1: Optimización de Proyectos	Pressman, R. S. (2021). <i>Ingeniería del software Un enfoque práctico</i> . McGraw-Hill.	X	
	Sommerville, I. (2011). <i>Ingeniería del software</i> . Pearson.	X	
	Weitzenfeld, A. (2008). <i>Ingeniería de Software Orientada a Objetos</i> . Cengage.	X	
	Harper-Smith, P., & Derry, S. (2012). <i>Administración de proyectos</i> . Trillas.		X
	Rodríguez, G. J. (2012, enero 15). Técnicas efectivas para la toma de requerimientos. Northware. https://www.northware.mx/blog/tecnicas-efectivas-para-la-toma-de-requerimientos/		X
	Software Guru (s/f). Lunch & Learn Webinar “¿Qué hacer para dominar el arte del levantamiento de requerimientos?” Youtube. Recuperado el 2 de octubre de 2024, de https://www.youtube.com/watch?v=FABsucursc0		X
	Scrum, L. (2020, septiembre 22). Técnicas que te ayudarán a Priorizar. Libre Scrum. https://librescrum.com/2020/09/22/tecnicas-de-priorizacion/		X
Landajueta, I. (s/f). Técnica de priorización MOSCOW. Blog personal. https://soka.gitlab.io/blog/post/2019-06-05-tecnicas-metodologias-priorizacion/		X	



Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Número y Nombre de la Unidad Didáctica	FORMATO APA	CLASIFICACIÓN	
		Básico	Consulta
Unidad 2: Restructuración de Proyectos	Blanes, R. G. (2018). <i>El libro práctico del programador ágil: Un Enfoque Integral Y Práctico Para El Desarrollo de Software Mediante Las Mejores Prácticas de Código Limpio, Refactoring, Testing, Principios de Diseño Y Gestión de la Configuración</i> . Independently Published.	X	
	Layton, M. C., Ostermiller, S. J., & Kynaston, D. J. (2020). <i>Agile project management for dummies</i> . John Wiley & Sons. Gómez,		X
	Weitzenfeld, A. (2005). <i>Ingeniería de software orientada a objetos con UML, Java e Internet</i> . Cengage Learning Editores.	X	
	Stevens, P., & Pooley, R. (2007). <i>Utilización de UML en ingeniería del software con objetos y componentes</i> . Addison Wesley.		X
	Norman, D. (2013). <i>The Design of Everyday Things: Revised and Expanded Edition</i> . Constellation.	X	
	Davis, C. (2023). <i>Domina Node.js: Desde principiantes hasta expertos</i> . Independently Published.	X	
	M. R. (2021). <i>Curso de desarrollo Web. HTML, CSS y JavaScript. Edición 2021</i> . Anaya multimedia	X	
	Wilkinson, C. (2020). <i>Ciencia de Datos Python: Una guía definitiva para que los principiantes aprendan los fundamentos de la ciencia de datos con Python</i> . Independently Published.	X	
	Cage, J. K. (2020). <i>Análisis de datos de Python para novatos: numpy/Pandas/Matplotlib/sklearn/Keras</i> .		X
	García, B. C. (2024). <i>Herramientas IA para impulsar tu productividad</i> . Ra-Ma Editorial.	X	
	Prada, R. (2023). <i>Integrando ChatGPT en tus aplicaciones Guía completa sobre la API de OpenAI</i> .	X	
	Alcántara, E. (2023). <i>Inteligencia Artificial Introducción: Para Novatos</i> . Independently Published.		X
	Monge, E. M. C., Monge, M. C., & Querol, J. L. (2022). <i>Temario oficial de informática: volumen III. Bases de datos avanzadas e ingeniería del software</i> . RA-MA	X	
	Fernández, G., Montero, J. (2024). <i>Fundamentos de ciencia de datos con R</i> . McGrawHill.		X
	Alexander, R. (2023). <i>Contar historias con datos</i> . McGrawHill.		X
	Pierson, L. (2017). <i>Data science for dummies</i> . John Wiley & Sons.	X	
Sweigart, A. (2021). <i>The Big Book of Small Python Projects: 81 Easy Practice Programs</i> . No Starch Press.		X	
Shan, C., Wang, H., Chen, W., & Song, M. (2015). <i>The Data Science Handbook: Advice and Insights from 25 Amazing Data Scientists</i> .		X	
Grus, J. (2015). <i>Data Science from Scratch</i> . O'Reilly Media.	X		





Programa Académico: Técnico en Programación

Unidad de Aprendizaje: Laboratorio de Desarrollo de Software IV

Número y Nombre de la Unidad Didáctica	FORMATO APA	CLASIFICACIÓN	
		Básico	Consulta
Unidad 3: Implementación de Proyectos	Guillen, G. (2018). <i>Ingeniería del Software un Enfoque Moderno</i> .	X	
	González, L. (2023). <i>Introducción a los Diferentes Tipos de Pruebas de Software</i> .		X
	Lastra, I. (2021). <i>Frontend testing</i> .	X	
	Ramos, K. (2020). <i>Estrategias de Localización de Selenium WebDriver: Guía Práctica para Potenciar las Pruebas Automatizadas</i> .	X	
	Jackvony, K. (2021). <i>The Complete Software Tester: Concepts, Skills, and Strategies for High-Quality Testing</i> .	X	
	Riera, M. F. (2023). <i>Puesta en producción segura</i> . Ra-Ma Editorial.	X	
	Martin, R. C. (2018). <i>Clean architecture: A Craftsman's Guide to Software Structure and Design</i> . Pearson Professional.	X	
	Ford, N., Parsons, R., & Kua, P. (2017). <i>Building evolutionary architectures: Support Constant Change</i> . O'Reilly Media.		X
García, A., Sanz, J. (2016). <i>Despliegue de aplicaciones web</i> . Garceta.		X	

