

 INSTITUTO SUPERIOR TECNOLÓGICO DEL AUSTRO Secretaría de Educación Superior, Ciencia, Tecnología e Innovación	VICERRECTORADO					PÁGINAS: 03		
	PLAN DE ESTUDIOS DE LA ASIGNATURA					VERSIÓN: 1		
	CÓDIGO: ISTAUSTRO-PA-PL-008					VIGENCIA DESDE: 27/03/2024		
CARRERA:								
DESARROLLO DE SOFTWARE								
COMPETENCIA GENERAL:								
Diseñar, desarrollar y mantener sistemas de software de calidad, que satisfagan las necesidades del mercado y cumplan con los estándares de calidad y seguridad requeridos en la industria tecnológica, mediante la aplicación de metodologías y técnicas adecuadas, el uso de tecnologías emergentes y el trabajo colaborativo en equipos interdisciplinarios, con una visión ética y responsable ante los desafíos y oportunidades del entorno global.								
I. DATOS DE LA ASIGNATURA								
ASIGNATURA	HORAS POR COMPONENTE			# DE CRÉDITOS	CODIGO ASIGNATURA	NIVEL	PARALELO	
	TOTAL (horas)	DOCENCIA	PRÁCTICA					AUTÓNOMO
Metodología de Desarrollo de Sistemas	144	64	48	32	3,00	DSN-AC1-204	II	A
PERIODO ACADÉMICO		# DE SEMANAS	UNIDAD DE ORGANIZACIÓN CURRICULAR		CAMPO DE FORMACIÓN		MODALIDAD	
IPAO-2025		16	Profesional		Adaptación e innovación tecnológica		Presencial	
ASIGNATURAS PRE-REQUISITO				ASIGNATURAS CO-REQUISITO				
CÓDIGO	ASIGNATURA			CÓDIGO	ASIGNATURA			
DSN-AC1-104	ANALISIS Y DISEÑO DE SISTEMAS							
II. INFORMACIÓN DEL DOCENTE								
NOMBRE		TÍTULO DE TERCER NIVEL		TÍTULO DE CUARTO NIVEL		CORREO ELECTRÓNICO		TELÉFONO
Cajamarca Ullauri Wilson Rolando		Ingeniero de Sistemas		Magister en Sistemas de Información Gerencial-MSIG		wilson.cajamarca@insteclrg.edu.ec		0984388910
III. DESCRIPCIÓN DE LA ASIGNATURA								
La asignatura Metodología de Desarrollo de Sistemas es de naturaleza teórico práctico. El propósito es desarrollar aplicaciones orientadas a satisfacer necesidades empresariales tecnológicas específicas, dentro de su formación profesional como Desarrollador de Software, para implementar software de calidad aplicando buenas prácticas de programación, así como adquirir habilidades de análisis y determinación de necesidades objetivas de las aplicaciones informáticas. El alumno será capaz de utilizar el contenido de esta asignatura con el fin de construir aplicaciones móviles con un nivel intermedio.								
IV. OBJETIVOS DE LA ASIGNATURA								
OBJETIVO GENERAL								
Desarrollar habilidades y conocimientos que permitan al estudiante planificar, diseñar, implementar y gestionar procesos de desarrollo de software mediante el uso de metodologías estructuradas y ágiles, asegurando la eficiencia, calidad y mantenibilidad de las aplicaciones informáticas								
OBJETIVOS ESPECÍFICOS								
Desarrollar la capacidad para aplicar metodologías estructuradas y ágiles en el diseño y desarrollo de software, garantizando la integración de buenas prácticas y estándares que favorezcan la calidad del código y la eficiencia en los procesos de desarrollo de software.								
Fomentar habilidades para gestionar el ciclo de vida completo del desarrollo de software, asegurando que las aplicaciones sean escalables, sostenibles y fácilmente mantenibles a lo largo del tiempo.								
V. CONTENIDOS DE LAS UNIDADES								
UNIDAD 1: METODOLOGÍA DE DESARROLLO TRADICIONALES								
OBJETIVO DE LA UNIDAD				PERFIL DE EGRESO				
El objetivo de la unidad 1 Metodologías de Desarrollo Tradicionales es comprender las metodologías de desarrollo de software tradicionales, sus características, ventajas y limitaciones, y cómo se aplican en el ciclo de vida del desarrollo de software.				Aplica metodologías tradicionales de desarrollo de software y justificar su uso en proyectos según las necesidades y características del mismo.				
#	RESULTADOS DE APRENDIZAJE							
1	Aplica técnicas de investigación en la búsqueda de nuevas formas de aplicación del desarrollo de software en los sectores industriales.							
2	Desarrolla trabajo independiente o colaborativo demostrando interés, creatividad, pensamiento crítico, ética, responsabilidad, empatía y liderazgo en las actividades asignadas.							
3	Valora la importancia del trabajo en equipo compartiendo conocimientos, problemas, soluciones, responsabilidades y toma de decisiones, respetando siempre la diversidad, gestión ambiental y fortaleciendo la tolerancia.							
4	Expresa las ideas con claridad y coherencia con una comunicación oral y escrita asertiva en las actividades asignadas.							
CÓDIGO	CONTENIDOS	COMPONENTES						
		DOCENCIA		PRÁCTICAS		AUTÓNOMO		
		HORAS	ACTIVIDAD	HORAS	ACTIVIDAD	HORAS	ACTIVIDAD	
1.1	Introducción	1	Charla docente participativa / Lluvia de ideas/ Aplicación de evaluación diagnóstica	0		0		
1.2	Generaciones de las metodologías de desarrollo de software	2	Aula invertida / Uso de material visual	0		8	TA1.1: Elaborar un documento que recopile la evolución de las generaciones de las metodologías de desarrollo de software; describir sus características, enfoques y avances con respecto a sus antecesoras o contemporáneas. Presente utilizando una herramienta informática para presentaciones en línea, sustentar el trabajo realizado.	
1.3	Ciclo de vida del desarrollo de software	2	Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0		

1.4	Principales metodologías convencionales	2	Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	
1.5	Metodologías de desarrollo de software SSADM	2	Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	
1.6	Metodologías de desarrollo de software MERISE	2	Charla docente participativa / Lluvia de ideas/ Uso de material visual	6	PR1.1: Analizar y establecer las diferencias entre las metodologías convencionales, SSADM y MERISE utilizando una herramienta de modelado o diagramación. Los estudiantes deberán comparar las metodologías convencionales, SSADM y MERISE, identificando sus diferencias clave en enfoque, estructura y aplicación. Para ello, utilizarán una herramienta informática como Microsoft Visio, Draw.io, Lucidchart o cualquier software de diagramación para representar visualmente las diferencias entre estas metodologías.	0	
1.7	Filosofía RUP	2	Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	
EV1.1	Evaluación de la unidad 1	1	Evaluación teórica	0		0	
TOTAL		14		6		8	
METODOLOGIA DIDÁCTICA		Aprendizaje por proyectos: Esta metodología consiste en plantear a los estudiantes un problema real que requiera el uso de la programación orientada a objetos para la creación de aplicaciones para dispositivos móviles. Los estudiantes deben trabajar de manera individual o en equipos para analizar el problema, identificar los objetos involucrados, definir las clases y sus relaciones e implementar una solución utilizando un lenguaje de programación. Charla participativa: Para cada uno de los temas de la unidad, el docente presentará los contenidos teóricos, ejemplos, y formará grupos de trabajo para generar un debate sobre los temas tratados.					
ESCENARIOS O AMBIENTES DE APRENDIZAJE		Aulas virtuales, bibliotecas virtuales, aulas físicas y biblioteca					
RECURSOS DIDÁCTICOS A UTILIZAR		Docente: Computador personal, textos bibliográficos y virtuales, esferos, material audiovisual, marcadores, software IDE para desarrollo de software. Estudiantes: Cuaderno de apuntes, lápices, esferos, computador personal, textos bibliográficos y virtuales, software IDE para desarrollo de software.					
UNIDAD 2: METODOLOGÍAS DE DESARROLLO ÁGILES							
OBJETIVO DE LA UNIDAD				PERFIL DE EGRESO			
El objetivo de la unidad 2 Metodologías de Desarrollo Ágiles es comprender y aplicar las principales metodologías ágiles en el desarrollo de software.				Aplica metodologías ágiles, como Scrum y Extreme Programming (XP), en proyectos de desarrollo de software que permita adaptarse a cambios y optimizar el trabajo en equipo y la satisfacción del cliente.			
#	RESULTADOS DE APRENDIZAJE						
1	Aplica conceptos, técnicas, herramientas de programación, que contribuyan con la implementación de soluciones de software.						
2	Desarrolla trabajo independiente o colaborativo demostrando interés, creatividad, pensamiento crítico, ética, responsabilidad, empatía y liderazgo en las actividades asignadas.						
3	Valora la importancia del trabajo en equipo compartiendo conocimientos, problemas, soluciones, responsabilidades y toma de decisiones, respetando siempre la diversidad, gestión ambiental y fortaleciendo la tolerancia.						
4	Expresa las ideas con claridad y coherencia con una comunicación oral y escrita asertiva en las actividades asignadas.						
CÓDIGO	CONTENIDOS	COMPONENTES					
		DOCENCIA		PRÁCTICAS		AUTÓNOMO	
		HORAS	ACTIVIDAD	HORAS	ACTIVIDAD	HORAS	ACTIVIDAD
2.1	Introducción a las metodologías ágiles	2	Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		8	TA2.1: Elaborar un documento que compare las metodologías ágiles con las metodologías tradicionales, identificando sus ventajas, desventajas y su aplicabilidad en distintos tipos de proyectos de desarrollo de software. Presente utilizando una herramienta informática para presentaciones en línea, sustentar el trabajo realizado.
2.2	Principios de Manifiesto Agile	2	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	
2.3	Metodología Cristal Clear	2	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	
2.4	Metodología SCRUM	6	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	10	PR2.1: Aplicación práctica de la metodología SCRUM en la gestión de un proyecto de software Se trabajarán en equipos para simular el uso de SCRUM en la gestión de un proyecto de software. Utilizarán una herramienta digital como JIRA, Trello o Monday.com u otra para organizar y visualizar las tareas del proyecto, aplicando los principios de SCRUM. Formar equipos de trabajo (4-5 estudiantes). Definir un proyecto sencillo (ejemplo: una aplicación de gestión de tareas u otro). Configurar el tablero SCRUM en JIRA, Trello o una herramienta similar. Asignar roles SCRUM: Scrum Master, Product Owner y Equipo de Desarrollo. Crear un backlog de producto con al menos 10 tareas. Planificar un sprint (1-2 semanas) con selección de tareas prioritarias. Realizar reuniones diarias simuladas y actualizar el estado de las tareas en la herramienta. Elaborar un informe final con capturas del tablero, descripción de la experiencia y aprendizajes.	0	
2.5	Metodología KANBAN	2	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	

2.6	Metodología Extreme Programming Xp	4	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	10	PR2.2: Simulación del desarrollo de un sistema de generación de certificados aplicando Extreme Programming (XP). Los estudiantes realizarán una simulación del proceso de desarrollo de un sistema de generación de certificados aplicando los principios de Extreme Programming (XP) sin necesidad de programar el sistema, enfocándose en la aplicación de la metodología. Formar equipos de trabajo (3-4 estudiantes) y definir los roles de XP: Programadores, Tester, Cliente (puede ser un miembro del equipo o el docente). Definir los requerimientos del sistema de generación de certificados mediante una reunión con el "cliente". Crear tarjetas CRC (Clase, Responsabilidad, Colaborador) para identificar los componentes clave del sistema. Simular el desarrollo con iteraciones cortas, aplicando: Pair Programming en papel o con pseudocódigo. Test-Driven Development (TDD) escribiendo pruebas antes de definir la solución. Refactorización, mejorando las soluciones en cada iteración. Realizar una retrospectiva del proceso y documentar los aprendizajes en un informe.	0	
2.7	Metodología de Desarrollo de APLICaciones Móviles	2	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	
EV2	Evaluación de la unidad 2	2	Evaluación teórica / práctica	0			
TOTAL		22		20		8	
METODOLOGIA DIDÁCTICA		Aprendizaje por proyectos: Esta metodología consiste en plantear a los estudiantes un problema real que requiera el uso de la programación orientada a objetos para la creación de aplicaciones para dispositivos móviles. Los estudiantes deben trabajar de manera individual o en equipos para analizar el problema, identificar los objetos involucrados, definir las clases y sus relaciones e implementar una solución utilizando un lenguaje de programación. Charla participativa: Para cada uno de los temas de la unidad, el docente presentará los contenidos teóricos, ejemplos, y formará grupos de trabajo para generar un debate sobre los temas tratados.					
ESCENARIOS O AMBIENTES DE APRENDIZAJE		Aulas virtuales, bibliotecas virtuales, aulas físicas y biblioteca					
RECURSOS DIDÁCTICOS A UTILIZAR		Docente: Computador personal, textos bibliográficos y virtuales, esferos, material audiovisual, marcadores, software IDE para desarrollo de software. Estudiantes: Cuaderno de apuntes, lápices, esferos, computador personal, textos bibliográficos y virtuales, software IDE para desarrollo de software.					
UNIDAD 3: UML: CASOS DE USO							
OBJETIVO DE LA UNIDAD				PERFIL DE EGRESO			
El objetivo de la Unidad 3 es modelar los requisitos de un sistema mediante diagramas de casos de uso en UML, y como estos facilitan la comunicación entre los involucrados en el desarrollo del software.				Crea y analiza diagramas de casos de uso en UML para representar de manera clara y detallada las interacciones entre los usuarios y el sistema			
#	RESULTADOS DE APRENDIZAJE						
1	Realiza procesos de análisis y verificación de consistencia de datos extraídos de diversas fuentes, que permitan generar reportes relevantes para una organización.						
4	Desarrolla trabajo independiente o colaborativo demostrando interés, creatividad, pensamiento crítico, ética, responsabilidad, empatía y liderazgo en las actividades asignadas.						
5	Valora la importancia del trabajo en equipo compartiendo conocimientos, problemas, soluciones, responsabilidades y toma de decisiones, respetando siempre la diversidad, gestión ambiental y fortaleciendo la tolerancia.						
6	Expresa las ideas con claridad y coherencia con una comunicación oral y escrita asertiva en las actividades asignadas.						
CÓDIGO	CONTENIDOS	COMPONENTES					
		DOCENCIA		PRÁCTICAS		AUTÓNOMO	
		HORAS	ACTIVIDAD	HORAS	ACTIVIDAD	HORAS	ACTIVIDAD
3.1	UML	2	Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		8	TA3.1: Investigación documental sobre UML y su aplicación en el desarrollo de software. Investigar y elaborar un documento que responda a las siguientes preguntas: ¿Qué es UML y cuál es su propósito en el desarrollo de software?. ¿Cuáles son los principales tipos de diagramas UML y en qué casos se utilizan?. ¿Cómo UML mejora la comunicación entre los equipos de desarrollo?. Presente utilizando una herramienta informática para presentaciones en línea, sustentar el trabajo realizado.
3.2	Diagramas de Caso de Uso	4	Aula Invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	
3.3	Relaciones	2	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		8	TA3.2: Investigación documental sobre diagramas de casos de uso y sus relaciones. Investigar y elaborar un documento que explique: ¿Qué es un diagrama de casos de uso y cuál es su propósito? ¿Cuáles son los principales elementos de un diagrama de casos de uso (actores, casos de uso, relaciones, sistema)?. ¿Cuáles son los tipos de relaciones en los casos de uso (inclusión, extensión y generalización) y cuándo se aplican?. Presente utilizando una herramienta informática para presentaciones en línea, sustentar el trabajo realizado.

3.4	Documentación de los Casos de Uso	4	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	10	PR3.1 Creación de Diagramas UML. Con el uso de herramientas como Lucidchart o Draw.io. Modelar un sistema básico de login donde el usuario ingresa su nombre de usuario y contraseña.Crear los Diagramas: Diagrama de Casos de Uso: Representa al Usuario interactuando con el sistema. Casos de uso: Ingresar credenciales, Validar, Acceder, Mostrar error. Diagrama de Clases: Define clases como Login, Usuario y Sistema de Validación. Diagrama de Secuencia: Muestra la interacción entre el Usuario y el Sistema de Validación. Entrega: Exportar los diagramas como imágenes (PNG o PDF) y envíalos por la plataforma de la asignatura.	0	
3.5	Diagramas de Estructura	2	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	
3.6	Diagramas de Comportamiento	1	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	
EVP1	Evaluación Parcial 1	1	Evaluación Teórico - Práctica	0		0	
3.7	Aplicación de una metodología de desarrollo de software según tendencia	4	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	0		0	
3.7,1	Desarrollo de un caso práctico aplicando una metodología ágil	6	Aula invertida / Charla docente participativa / Lluvia de ideas/ Uso de material visual	12	PR3.2: Desarrollo de Casos de Uso para un Sistema de Generación de Certificados Aplicando Metodología Ágil. Desarrollar casos de uso para un sistema de generación de certificados u otro, aplicando una metodología ágil como Scrum o Extreme Programming (XP). Los estudiantes deben documentar los casos de uso, los diagramas asociados y aplicar la metodología para planificar y gestionar el desarrollo del sistema. Para ello elegir una metodología ágil (Scrum o XP) y trabaja en equipos de 3 a 4 estudiantes.Definir los requisitos del sistema de generación de certificados (actores, información, formato, generación, etc.). Documentar al menos 3 casos de uso. Realiza un diagrama de casos de uso.	0	
EV3	Evaluación de la unidad 3	1	Evaluación teórica - práctica	0		0	
EVP2	Evaluación parcial 2	1	Evaluación teórica / práctica	0		0	
TOTAL		28		22		16	

METODOLOGIA DIDÁCTICA	Aprendizaje por proyectos: Esta metodología consiste en plantear a los estudiantes un problema real que requiera el uso de la programación orientada a objetos para la creación de aplicaciones para dispositivos móviles. Los estudiantes deben trabajar de manera individual o en equipos para analizar el problema, identificar los objetos involucrados, definir las clases y sus relaciones e implementar una solución utilizando un lenguaje de programación. Charla participativa: Para cada uno de los temas de la unidad, el docente presentará los contenidos teóricos, ejemplos, y formará grupos de trabajo para generar un debate sobre los temas tratados.
ESCENARIOS O AMBIENTES DE APRENDIZAJE	Aulas virtuales, bibliotecas virtuales, aulas físicas y biblioteca
RECURSOS DIDÁCTICOS A UTILIZAR	Docente: Computador personal, textos bibliográficos y virtuales, esferos, material audiovisual, marcadores, software IDE para desarrollo de software. Estudiantes: Cuaderno de apuntes, lápices, esferos, computador personal, textos bibliográficos y virtuales, software IDE para desarrollo de software.

VI. SISTEMA DE EVALUACIÓN

Componente		Peso %	Estrategia de evaluación
Docencia	Parcial I	20%	Se evalúa la entrega puntual y completa de trabajos y tareas con calidad académica, que demuestren el desempeño y el compromiso del estudiante, lecciones orales o escritas, presenciales y/o virtuales de acuerdo al tema y planificación de unidad.
	Parcial II	20%	
Práctico	Parcial I	30%	Se evalúa la ejecución de talleres, prácticas de laboratorio, visitas técnicas, charlas técnicas, casos prácticos, talleres de ejercicios de acuerdo al tema y planificación de unidad.
	Parcial II	30%	
Trabajo Autónomo	Parcial I	30%	Comprende el trabajo realizado por el estudiante, orientado al desarrollo de capacidades para el aprendizaje individual y/o grupal. Este trabajo será diseñado, planificado y orientado por el profesor para alcanzar los objetivos y el perfil de egreso/competencia general de la carrera.
	Parcial II	30%	
Evaluación Parcial I		20%	Es el resultado de aplicar el instrumento de evaluación, previamente aprobado por la Coordinación de Carrera. Este instrumento abordará el total de los contenidos correspondientes al parcial I.
Evaluación Parcial II		20%	Es el resultado de aplicar el instrumento de evaluación, previamente aprobado por la Coordinación de Carrera. Este instrumento abordará el total de los contenidos correspondientes al parcial II.
Total		100%	En cada parcial/ la nota final considera el promedio de los dos parciales.

La calificación final de aprobación será el promedio de los dos parciales sobre 10 puntos cada uno. Cada estudiante deberá obtener una calificación mínima de 7 puntos (equivalente al 70%) en todas las asignaturas para ser promovido al siguiente ciclo académico. En caso de obtener menos de 7 puntos, en el promedio final, la o el estudiante reprueba la asignatura.

VII. BIBLIOGRAFÍA

6.1 BÁSICA: la selección de estos documentos como parte de la bibliografía de la asignatura sirve para reforzar y consolidar los conocimientos asociados con el logro de objetivos cognitivos y procedimentales de los estudiantes.

#	Autor(es)	Título	Código biblioteca/URL	Año	Editorial
---	-----------	--------	-----------------------	-----	-----------

1	Abuchar Porras, Alexandra	Metodologías ágiles para el desarrollo de software	https://books.google.com.ec/books/about/Metodolog%C3%ADas_%C3%A1giles_para_el_desarrollo.html?id=JfXB_EAAQBAJ&redir_esc=y	2023	Editorial Universidad Distrital Francisco José de Caldas
2	Scrumstudy	Scrum Fundamentals Certified	https://www.scrumstudy.com/certification/scrum-fundamentals-certified	2022	
6.2 COMPLEMENTARIA Y DE CONSULTA: la selección de la bibliografía complementaria y de consulta estan asociados a la creación de habilidades de pensamiento genérico del estudiante.					
#	Autor(es)	TÍTULO	Código biblioteca/URL	Año	Editorial
1	PEARSON ADISSON WESLEY	INGENIERIA DE SOFTWARE	ISTA-005.1-004	2005	SOMERVILLE
6.2.1 PÁGINAS WEB					
#	PARÁMETRO DE BÚSQUEDA	TÍTULO	URL	AUTOR	FECHA
1	metodologias de desarrollo de software	Metodologías de Desarrollo de Software: guía para optimizar tu proyecto	https://es.abstracta.us/blog/metodologias-de-desarrollo-de-software-guia-completa/	Abstracta	2024
2	scrum	The Home to Scrum	https://www.scrum.org/	Scrum org	2024
3	metodologias agiles de desarrollo de software	Metodologías Ágiles: Qué son y cuales son las más utilizadas	https://www.iebschool.com/hub/que-son-metodologias-agiles-agile-scrum/	IEBS	2024
VIII. FIRMAS					
Elaborado por:		Revisado por:		Aprobado por:	
Ing. Wilson Cajamarca, Msc. DOCENTE		Ing. Cristian Cáceres, Msc. COORDINADOR DE CARRERA		Ing. Gustavo Guillén, Mgs. VICERRECTOR	
Fecha: 27/03/2025		Fecha: 27/03/2025		Fecha: 27/03/2025	