

SÍLABO ARQUITECTURA DE SOFTWARE PARA SISTEMAS DE INFORMACIÓN

ÁREA CURRICULAR: SISTEMAS DE INFORMACION



I. DATOS GENERALES

Departamento Académico	Ingeniería y Arquitectura		
Escuela Profesional	Ingeniería de Computación y Sistemas		
Programa académico	Ingeniería de Computación y Sistemas		
Semestre Académico	2025-I		
Tipo de asignatura	Obligatoria		
Tipo de estudios	General ()	Específica ()	Especialidad (X)
Modalidad de la asignatura	Presencial (X)	Semipresencial ()	A distancia ()
Código de la asignatura	09140507021		
Ciclo	VII		
Requisitos	09011906052 – Ingeniería De Software I		
Cantidad de horas	Teoría (00) Práctica (04) Total horas (04)		
	Teoría lectiva presencial ()		
	Teoría lectiva a distancia ()		
	Teoría no lectiva presencial ()		
	Teoría no lectiva a distancia ()		
	Práctica lectiva presencial (04)		
	Práctica lectiva a distancia ()		
	Práctica no lectiva presencial ()		
Cantidad de Créditos	Práctica no lectiva a distancia ()		
	Teoría (00)	Práctica (02)	Total créditos (02)
Docente(s)	Dr. Huanco Ramos, Fidel		

II. SUMILLA

Es de naturaleza formación de especialidad, tiene el propósito de potenciar en el estudiante sus habilidades para analizar y diseñar una arquitectura de software para los sistemas de información. Se desarrolla los siguientes contenidos: Introducción a la arquitectura de software, vistas y estilos de la arquitectura, requisitos de calidad de un software, diagramación arquitectónica de software, patrones de arquitectura, arquitectura orientada a servicios, Arquitecturas en Cloud Computing, Arquitecturas para software en dispositivos móviles y documentación de una arquitectura de software.

La asignatura exige del estudiante la elaboración de un trabajo integrador.

III. COMPETENCIAS Y SUS COMPONENTES COMPRENDIDOS EN LA ASIGNATURA

3.1 Competencias:

- Analizar un sistema complejo de computación y aplicar principios de computación y otras disciplinas relevantes para identificar soluciones. (1)
- Diseñar, implementar y evaluar una solución basada en computación para cumplir con un determinado conjunto de requerimientos de computación en el contexto de la disciplina del programa. (2)
- Comunicación efectiva en una variedad de contextos profesionales. (3)
- Reconocer la responsabilidad profesional y realizar juicios informados en la práctica de computación basados en principios éticos y legales. (4)
- Trabajar de manera efectiva como miembro o líder de un equipo comprometido con actividades propias de la disciplina del programa. (5)
- Brindar soporte a la entrega, uso, y administración de sistemas de información en un entorno de sistemas de información. (6)

3.2. Componentes

Capacidades:

- Aplica conocimiento de la especialidad (1.a)
- Analiza problemas y establece requerimientos (1.b)
- Diseña, implementa y/o evalúa procesos, componentes o programas (2.a)
- Usa técnicas y herramientas de la especialidad (2.b)
- Se comunica de manera efectiva (3.a)
- Entiende aspectos de responsabilidad profesional, ética y social (4.a)
- Analiza el impacto de su solución en la organización y la sociedad (4.b)
- Trabaja en equipo (5.a)
- Participa en las diversas etapas de un sistema de información (6.a)

Contenidos actitudinales

- Búsqueda de la verdad. (b)
- Compromiso ético en todo su quehacer. (c)
- Integridad (honestidad, equidad, justicia, solidaridad y vocación de servicio). (d)
- Compromiso con la calidad y búsqueda permanente de la excelencia (e)

IV. PROGRAMACIÓN DE CONTENIDOS



UNIDAD I. INTRODUCCION A LA ARQUITECTURA DE SOFTWARE

CAPACIDAD:

Aplica conocimiento de la especialidad, Analiza problemas y establece requerimientos, Diseña, implementa y/o evalúa procesos, componentes o programas, Usa técnicas y herramientas de la especialidad, Se comunica de manera efectiva.

Entiende aspectos de responsabilidad profesional, ética y social

Analiza el impacto de su solución en la organización y la sociedad, Trabaja en equipo y Participa en las diversas etapas de un sistema de información

SEMANA	CONTENIDOS CONCEPTUALES	CONTENIDOS PROCEDIMENTALES	ACTIVIDADES DE APRENDIZAJE	HORAS LECTIVAS		HORAS NO LECTIVAS	
				TEORÍA	PRÁCTICA	TEORÍA	PRÁCTICA
1	Que es la arquitectura de software, Que procesos soportan la arquitectura de software, flujos de trabajo de la arquitectura de software	<ul style="list-style-type: none"> . Reconoce los conceptos de arquitectura software . Describe los procesos y los flujos de trabajo 	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica	0	4	0	0
			<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica				
2	Patrones, Practicas y Principios de la Arquitectura de Software, Niveles de abstracción arquitecturales, Que es mala o buena arquitectura	<ul style="list-style-type: none"> . Reconoce los patrones, practicas y principios de la arquitectura de software 	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica	0	4	0	0
			<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica				
3	Arquitectura Centrada en Dominios, Tipos de Arquitectura Centrada en Dominios, ventajas y desventajas, Arquitectura Centrada en base de datos vs centrada en dominios, tipos de arquitectura.	<ul style="list-style-type: none"> - Reconoce la arquitectura centrada en dominios sus ventajas y desventajas - Reconoce la ventajas y desventajas de una arquitectura centrada en base datos. 	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica	0	4	0	0
			<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica				



UNIDAD II. DISEÑO DE LA ARQUITECTURA DE SOFTWARE							
CAPACIDAD:							
SEMANA	CONTENIDOS CONCEPTUALES	CONTENIDOS PROCEDIMENTALES	ACTIVIDADES DE APRENDIZAJE	HORAS LECTIVAS		HORAS NO LECTIVAS	
				TEORÍA	PRÁCTICA	TEORÍA	PRÁCTICA
4	Capas de una Aplicación, Inversión de Dependencias, Porque usar capas en una aplicación, Cuales son las Capas de una aplicación	-Reconoce las capas de una aplicación, así como usa la inversión de dependencia. -Diferencia los pros y los cons de usar capas para una aplicación	<u>Lectivas:</u> Desarrollo del tema – 2 horas · Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> · No aplica	0	4	0	0
5	Diseño de la Arquitectura usando el Modelo C4, Arquitectura Monolitica	-Reconoce el modelo C4 de Simon Brown -Reconoce cuando usar una arquitectura monolitica	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica				
6	Organización Funcional, Organización Categórica -	-Reconoce la importancia de una arquitectura Funcional y su organización, así como se aplica una arquitectura categórica	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica	0	4	0	0
7	Screaming Architecture, Organización Funcional vs categórica, ventajas y desventajas -	-Resume la importancia del screaming de la arquitectura dentro de la organización funcional vs categorica	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica				
8	Examen Parcial						

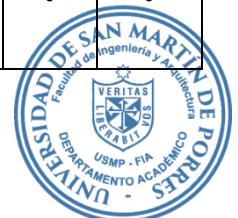


UNIDAD III. ARQUITECTURA DISTRIBUIDA PARA LA ESCALIBILIDAD

CAPACIDAD:

- Aplica conocimiento de la especialidad, Analiza problemas y establece requerimientos, Diseña, implementa y/o evalúa procesos, componentes o programas,
- Usa técnicas y herramientas de la especialidad, Se comunica de manera efectiva, Entiende aspectos de responsabilidad profesional, ética y social
- Analiza el impacto de su solución en la organización y la sociedad, Trabaja en equipo y Participa en las diversas etapas de un sistema de información

SEMANA	CONTENIDOS CONCEPTUALES	CONTENIDOS PROCEDIMENTALES	ACTIVIDADES DE APRENDIZAJE	HORAS LECTIVAS		HORAS NO LECTIVAS	
				TEORÍA	PRÁCTICA	TEORÍA	PRÁCTICA
9	Microservicios, Límites del Desacoplamiento, Ventajas y desventajas	<ul style="list-style-type: none"> - Reconoce la importancia de los microservicios y su relación con la arquitectura centrada en dominios, sus límites, ventajas y desventajas 	<u>Lectivas:</u> Desarrollo del tema – 2 horas <ul style="list-style-type: none"> · Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> <ul style="list-style-type: none"> · No aplica 	0	4	0	0
10	Arquitectura Orientada a Eventos, Ventajas y desventajas	<ul style="list-style-type: none"> - Reconoce los componentes de la arquitectura orientada a eventos, los tópicos y colas, sus ventajas en la forma asíncrona y sincrónica. 	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica	0	4	0	0
11	Mantenibilidad y Pruebas de las Arquitecturas distribuidas	<ul style="list-style-type: none"> - Reconoce la facilidad de la mantenibilidad de la arquitectura distribuida 	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica	0	4	0	0
12	Casos Prácticos de Arquitectura de Software (monolítica vs Distribuida)	<ul style="list-style-type: none"> - Empieza a trabajar con casos prácticos de arquitectura distribuida y monolítica 	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas <u>De trabajo Independiente</u> No aplica	0	4	0	0



UNIDAD IV. PATRONES AVANZADOS DE ARQUITECTURA

CAPACIDAD:

- Aplica conocimiento de la especialidad, Analiza problemas y establece requerimientos, Diseña, implementa y/o evalúa procesos, componentes o programas,
- Usa técnicas y herramientas de la especialidad, Se comunica de manera efectiva, Entiende aspectos de responsabilidad profesional, ética y social
- Analiza el impacto de su solución en la organización y la sociedad, Trabaja en equipo y Participa en las diversas etapas de un sistema de información

SEMANA	CONTENIDOS CONCEPTUALES	CONTENIDOS PROCEDIMENTALES	ACTIVIDADES DE APRENDIZAJE	HORAS LECTIVAS		HORAS NO LECTIVAS						
				TEORÍA	PRÁCTICA	TEORÍA	PRÁCTICA					
13	Patrones de Microservicios, Detalles de uso en la Arquitectura	<ul style="list-style-type: none"> - Reconoce los patrones de arquitectura de microservicios 	<u>Lectivas:</u> Desarrollo del tema – 2 horas <ul style="list-style-type: none"> · Ejercicios guiados – 2 horas 	0	4	0	0					
			<u>De trabajo Independiente</u> <ul style="list-style-type: none"> · No aplica 									
14	Patrones de Arquitectura de Componentes, Detalles de Arquitectura GOF	<ul style="list-style-type: none"> - Reconoce los patrones GOF ventajas y desventajas 	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas	0	4	0	0					
			<u>De trabajo Independiente</u> No aplica									
15	Sustentación del trabajo final: Selección de los mejores trabajos	<ul style="list-style-type: none"> - Resume todo el trabajo en un caso práctico expuesto en clase. 	<u>Lectivas:</u> Desarrollo del tema – 2 horas Ejercicios guiados – 2 horas	0	4	0	0					
			<u>De trabajo Independiente</u> No aplica									
16	Examen final											
17	Entrega de promedios finales											



V. ESTRATEGIAS DIDÁCTICAS

- **Método Expositivo – Interactivo.** Comprende la exposición del docente y la interacción con el estudiante, empleando las herramientas disponibles en el aula virtual de la asignatura
- **Método de Discusión Guiada.** Conducción del grupo para abordar situaciones y llegar a conclusiones y recomendaciones, empleando las herramientas disponibles en el aula virtual de la asignatura
- **Método de Demostración – Ejecución.** Se utiliza para ejecutar, demostrar, practicar y retroalimentar lo expuesto, empleando las herramientas disponibles en el aula virtual de la asignatura

VI. RECURSOS DIDÁCTICOS

Equipos: computadora, ecran, proyector de multimedia.

Materiales: Material docente, pizarra, prácticas dirigidas de laboratorio, videos tutoriales, foros y textos bases (ver fuentes de consultas).

Software: Vmware Client, Windows Server, Packet Tracer

VII. EVALUACIÓN DEL APRENDIZAJE

El promedio final de la asignatura se obtiene con la siguiente fórmula:

$$\begin{aligned} \text{PF} &= (2*\text{PE}+\text{EP}+\text{EF})/4 \\ \text{PE} &= ((\text{P1}+\text{P2}+\text{P3}+\text{P4}-\text{MN})/3 + \text{W1}) /2 \end{aligned}$$

Donde:

PF = Promedio final
PE = Promedio de evaluaciones
EP = Examen parcial
EF = Examen final
P1 - P4 = Evaluaciones.
W1 = Trabajo
MN= Menor nota

VIII. FUENTES DE INFORMACIÓN.

- Mark Richards; Neal Ford (2020), Fundamentals of Software Architecture
- Chris Richardson (2018), Microservices Patterns
- Robert C. Martin (2017), Clean Architecture: A Craftsman's Guide to Software Structure and Design
- Mark Richards (2015), Software Architecture Patterns
- Morgan Kaufmann (2015), Continuous Architecture

IX. APORTE DE LA ASIGNATURA AL LOGRO DE RESULTADOS

El aporte de la asignatura al logro de los Resultados del Estudiante (Student Outcomes) en la formación del graduado en Ingeniería de Computación y Sistemas, se establece en la tabla siguiente:

K = clave R = relacionado Recuadro vacío = no aplica

1	Analizar un sistema complejo de computación y aplicar principios de computación y otras disciplinas relevantes para identificar soluciones.	K
2	Diseñar, implementar y evaluar una solución basada en computación para cumplir con un determinado conjunto de requerimientos de computación en el contexto de la disciplina del programa.	K
3	Comunicación efectiva en una variedad de contextos profesionales.	R



4	Reconocer la responsabilidad profesional y realizar juicios informados en la práctica de computación basados en principios éticos y legales.	K
5	Trabajar de manera efectiva como miembro o líder de un equipo comprometido con actividades propias de la disciplina del programa.	R
6	Brindar soporte a la entrega, uso, y administración de sistemas de información en un entorno de sistemas de información	R
7	Aprendizaje Continuo: Reconoce la necesidad y tiene la capacidad para dedicarse a un aprendizaje autónomo para el desarrollo profesional continuo.	

