

SÍLABO

Diseño y arquitectura de software (100000S03I)

2024 - Ciclo 2 Agosto

1. DATOS GENERALES

1.1. Carrera:	Ingeniería de Sistemas e Informática
1.2. Créditos:	3
1.3. Enseñanza de curso:	Presencial
1.4. Horas semanales:	4

2. FUNDAMENTACIÓN

La presente asignatura le brindará al estudiante un panorama amplio sobre lo que significa diseño y arquitectura de software considerando los siguientes aspectos: problemas, principios, patrones, técnicas, métodos, optimización de caché, wrappers, inversión de control, programación orientada a aspectos, balanceo de carga, frameworks, orquestación y coreografía de servicios web. Asimismo brindará al estudiante sólidos conocimientos acerca de arquitecturas como SOA, MVC, REST y arquitecturas Cloud Computing.

3. SUMILLA

La asignatura es de carácter teórico-práctico. Se desarrolla los siguientes contenidos: Introducción a la arquitectura de software, vistas y estilos de la arquitectura, requisitos de calidad de un software, diagramación UML orientada al diseño arquitectónico de software, patrones de arquitectura, arquitectura orientada a servicios (SOA), Arquitecturas en Cloud Computing, Arquitecturas para software en dispositivos móviles y documentación de una arquitectura de software.

4. LOGRO GENERAL DE APRENDIZAJE

Al finalizar el curso, el estudiante elabora un diseño arquitectónico basado en los requerimientos fundamentales del producto, considerando la relevancia de la arquitectura orientada a servicios (SOA).

5. UNIDADES Y LOGROS ESPECÍFICOS DE APRENDIZAJE

Unidad de aprendizaje 1: Introducción a la Arquitectura de Software.	Semana 1,2,3 y 4
Logro específico de aprendizaje: El estudiante formula un informe descriptivo sobre los requerimientos de un producto de software para las necesidades de los Stakeholders, aplicando un lenguaje de modelado como UML y un proceso de desarrollo en la herramienta IBM RSA; definiendo la Visión del Negocio, modelo de casos de Uso y prototipos visuales.	
Temario: <ul style="list-style-type: none"> Introducción al diseño y la arquitectura de software. Estado del Arte de la Arquitectura de Software. Contexto del diseño en múltiples ciclos de vida del desarrollo de software. Mejores Prácticas de la Ing. de Software. Proceso de desarrollo. Documento de Visión. Principios y procesos en las arquitecturas de software. Principios del Diseño. Acoplamiento y cohesión. Conceptos de Orientación a Objetos. Requerimientos de Software. Documento Especificación De requerimientos de software (SRS). Diagramas UML orientados a la arquitectura de software. Modelo de Casos de Uso. Modelo del Dominio. Diagramas de Componentes. Diagramas de Despliegue. Visión de Arquitecturas de software existente. Arquitecturas Cliente Servidor. Arquitecturas Distribuidas. Arquitecturas MVC. Arquitecturas SOA. Arquitecturas REST. Arquitecturas Cloud Computing. 	
Unidad de aprendizaje 2: Diseño de Arquitecturas de Software.	Semana 5,6,7,8 y 9

Logro específico de aprendizaje:

Al finalizar la segunda unidad, el estudiante diseña la arquitectura de un producto de software, haciendo uso de patrones arquitectónicos, elementos del modelo de diseño, clases de diseño, subsistemas, interfaces en la herramienta IBM RSA; cumpliendo con los requerimientos fundamentales del software a construir.

Temario:

- Análisis de la arquitectura de software. Análisis de Arquitecturas de Software. Patrones Arquitectónicos. Análisis de arquitecturas centradas en casos de uso.
- Diseño de la arquitectura de software. Elementos de Diseño. Mecanismos de Diseño. Patrones de Diseño.
- Arquitecturas de software en Tiempo de Ejecución. Modelo de distribución. Diagramas de Interacción. Diagramas de Componentes. Diagramas de Despliegue. GUI's.
- Patrones y Antipatrones en las arquitecturas de software. Optimización de caché. Wrappers. Inversión de control e Inyección de Dependencias. Programación Orientada a Aspectos. Balanceo de carga. Frameworks. Método ADD (Attribute Driven Design).
- Análisis y Diseño de patrones arquitectónicos de software. Tácticas para el diseño de Arquitecturas de software. Modelos de diseño. Diagramas de secuencia de Diseño arquitectónico

Unidad de aprendizaje 3:

Diseño detallado de Arquitecturas de Software.

Semana 10,11,12,13,14,15,16,17 y 18

Logro específico de aprendizaje:

Al finalizar la tercera unidad, el estudiante diseña componentes y clases de software; haciendo uso de la herramienta IBM RSA, patrones de diseño que garanticen la integración, en el marco de una arquitectura definida, cumpliendo con los requerimientos fundamentales del software a construir.

Temario:

- Análisis Detallado y Diseño Detallado de Arquitecturas de software. Diseño de Subsistemas. Diagramas de subsistemas.
- Diseño Arquitectónico basado en UML para subsistemas. Diseño de Clases y Sub Clases. Diagramas de Clases.
- Diseño Arquitectónico basado en UML para la integración de subsistemas. Atributos y Subatributos de Calidad de Software. Contratos de Servicios.
- Diseño Arquitectónico basado en UML para subsistemas en SOA. Protocolos SOA. Orquestación y Coreografía de servicios web. Diagramas de Componentes para SOA. Diagramas de Despliegue para SOA.
- Diseño Arquitectónico basado en UML para subsistemas en Arquitecturas para software de dispositivos móviles. Protocolos para software para dispositivos móviles. Diagramas de Componentes para REST. Diagramas de Despliegue para REST.
- Arquitecturas Cloud Computing para software en dispositivos móviles.
- Herramienta IBM RSA.
- Generación de Arquitecturas de servicios utilizando Azure SAS.

6. METODOLOGÍA

Para el desarrollo de los aprendizajes del curso, una de las estrategias que se propone es la exposición del docente que proporciona la construcción de los conocimientos a partir de ejemplos y casuísticas que faciliten la comprensión. Asimismo, se promueve la participación activa y permanente del estudiante a través del desarrollo de ejercicios, lecturas, absolución de preguntas, en forma individual y grupal (aprendizaje colaborativo) lo que permite un trabajo metacognitivo, a través de la actividad autónoma del estudiante en el desarrollo de las evaluaciones del curso (aprendizaje autónomo). Por ello es importante que el estudiante asista a las clases, habiendo leído los temas correspondientes a cada sesión. Finalmente, se utilizan otros recursos, como: pizarra, multimedia, videos (aprendizaje para la era digital) y comunicación a través de medios complementarios como correos electrónicos para fomentar una mayor interacción con el estudiante.

7. SISTEMA DE EVALUACIÓN

El cálculo del promedio final se hará de la siguiente manera:

$$(20\%)PC1 + (20\%)PC2 + (30\%)PC3 + (30\%)EXFI$$

Donde:

Tipo	Descripción	Semana	Observación
PC1	PRÁCTICA CALIFICADA 1	5	
PC2	PRÁCTICA CALIFICADA 2	10	
PC3	PRÁCTICA CALIFICADA 3	15	

Tipo	Descripción	Semana	Observación
EXFI	EXAMEN FINAL INDIVIDUAL	18	

Indicaciones sobre Fórmulas de Evaluación:

1. La nota mínima aprobatoria final es de 12.
2. El estudiante que no rinde el examen final puede rendir un único examen de rezagado. La nota obtenida en este examen de rezagado reemplaza al examen final no rendido.
El estudiante rinde el examen de rezagado en la fecha programada por la Universidad, previa presentación de solicitud y pago de los derechos por examen de rezagado dispuesto en el tarifario vigente y publicado en Portal del Estudiante. Los exámenes de rezagados se aplican al final del período lectivo y abarcan todos los temas vistos en la asignatura.
3. En caso un estudiante no rinda una práctica calificada (PC) y, por lo tanto, obtenga NSP, este es reemplazado por la nota obtenida en el examen final. Si también tiene NSP en el examen final, este es reemplazado por la nota obtenida en el examen rezagado. Este reemplazo de nota es automático. No es necesario que el estudiante realice trámite alguno para que proceda el remplazo de la nota. En caso de que el alumno tenga más de una práctica calificada no rendida, solo se reemplaza la práctica calificada de mayor peso.

8. FUENTES DE INFORMACIÓN

Bibliografía Base:

- Ivana Harari ; Díaz, Javier ; Ana Paola Amadeo. (2013). *Guía de recomendaciones para diseño de software centrado en el usuario*. D - Editorial de la Universidad Nacional de La Plata. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=36892>
- Luis, C., Perla, V., & Humberto, C. . *Arquitectura del Software*. Cengage Learning Editores SA. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=29509>

Bibliografía Complementaria:

- García Mireles, Gabriel Alberto. *Aplicación del modelado de procesos en un curso de ingeniería de software*. Red Revista Electrónica de Investigación Educativa. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=35844>
- Sosa Flores, Miguel. *Diseño de software para evaluación de inversiones en tecnologías de información y comunicación*. El Cid Editor. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=36378>
- Segovia Pérez, Javier. (2005). *Introducción a la Ingeniería del Software: modelos de desarrollo de programas*. Delta Publicaciones. <https://tubiblioteca.utp.edu.pe/cgi-bin/koha/opac-detail.pl?biblionumber=37015>

9. COMPETENCIAS

Carrera	Competencias específicas
Ingeniería de Sistemas e Informática	<ul style="list-style-type: none"> • Gestión de Sistemas de Información • Análisis de Sistemas

10. CRONOGRAMA DE ACTIVIDADES

Unidad de aprendizaje	Semana	Sesión	Tema	Actividades y evaluaciones
	1	1 - 2	Introducción al diseño y la arquitectura de software. Estado del Arte de la Arquitectura de Software. Contexto del diseño en múltiples ciclos de vida del desarrollo de software. Mejores Prácticas de la Ing. de Software. Proceso de desarrollo. Documento de Visión.	<ul style="list-style-type: none"> • Prueba de Entrada • Analiza documentos referentes al Diseño en la Ingeniería de Software.
			Principios y procesos en las arquitecturas de software. Principios del Diseño. Acoplamiento y cohesión. Conceptos de Orientación a Objetos. Requerimientos de Software. Documento Especificación De requerimientos de software (SRS).	<ul style="list-style-type: none"> • Elabora una descripción global del sistema, especifica requerimientos funcionales y no

Unidad 1 Introducción a la Arquitectura de Software	2	3 - 4		funcionales del sistema en herramienta IBM RSA. Desarrolla en equipo documento de Visión y SRS de un caso práctico en la herramienta IBM RSA.
	3	5 - 6	Diagramas UML orientados a la arquitectura de software. Modelo de Casos de Uso. Modelo del Dominio. Diagramas de Componentes. Diagramas de Despliegue.	<ul style="list-style-type: none"> Elabora el modelado arquitectura centrado en casos de uso, un modelo de dominio, diagramas de componentes diagramas de despliegue utilizando la herramienta IBM RS.
	4	7 - 8	Visión de Arquitecturas de software existente. Arquitecturas Cliente Servidor. Arquitecturas Distribuidas. Arquitecturas MVC. Arquitecturas SOA. Arquitecturas REST. Arquitecturas Cloud Computing.	<ul style="list-style-type: none"> Elabora una arquitectura cliente servidor
Unidad 2 Diseño de Arquitecturas de Software	5	9 - 10	Análisis de la arquitectura de software. Análisis de Arquitecturas de Software. Patrones Arquitectónicos. Análisis de arquitecturas centradas en casos de uso.	<ul style="list-style-type: none"> Identificación de paquetes por capas, existiendo trazabilidad entre el modelo de diseño y el modelo de implementación usando la herramienta IBM RSA.
			Evaluación	<ul style="list-style-type: none"> PRÁCTICA CALIFICADA 1
	6	11 - 12	Diseño de la arquitectura de software. Elementos de Diseño. Mecanismos de Diseño. Patrones de Diseño.	<ul style="list-style-type: none"> Establece la relación entre especificación de Casos de Uso y las operaciones del sistema, existiendo trazabilidad entre ellos.
	7	13 - 14	Arquitecturas de software en Tiempo de Ejecución. Modelo de distribución. Diagramas de Interacción. Diagramas de Componentes. Diagramas de Despliegue. GUI's.	<ul style="list-style-type: none"> Desarrolla una Estructura de Diagramas de interacción, de componentes y de despliegue existiendo trazabilidad entre las Especificaciones de casos de uso modelo de análisis y prototipos de las GUI's.

Unidad 3 Diseño detallado de Arquitecturas de Software	8	15 - 16	Patrones y Antipatrones en las arquitecturas de software. Optimización de caché. Wrappers. Inversión de control e Inyección de Dependencias. Programación Orientada a Aspectos. Balanceo de carga. Frameworks. Método ADD (Attribute Driven Design).	<ul style="list-style-type: none"> Elabora programas orientados a aspectos y balance de cargas.
	9	17 - 18	Análisis y Diseño de patrones arquitectónicos de software. Tácticas para el diseño de Arquitecturas de software. Modelos de diseño. Diagramas de secuencia de Diseño arquitectónico	<ul style="list-style-type: none"> Elabora un diagrama de secuencia de diseño para un caso de uso de un problema planteado. Utiliza la herramienta IBM RSA para elaborar diagramas de interacción.
	10	19 - 20	Análisis Detallado y Diseño Detallado de Arquitecturas de software. Diseño de Subsistemas. Diagramas de subsistemas.	<ul style="list-style-type: none"> Presenta la estructuración de los subsistemas, existiendo trazabilidad entre el modelo de análisis, el modelo de diseño propuesto y la implementación del mismo. Utiliza la herramienta IBM RSA
			Evaluación	<ul style="list-style-type: none"> PRÁCTICA CALIFICADA 2
	11	21 - 22	Diseño Arquitectónico basado en UML para subsistemas. Diseño de Clases y Sub Clases. Diagramas de Clases.	<ul style="list-style-type: none"> Desarrolla criterios del diseño de las clases para casos de uso, especificando los parámetros, reglas, condiciones usando la misma sintaxis del lenguaje de programación seleccionado. Utiliza la herramienta IBM RSA.
	12	23 - 24	Diseño Arquitectónico basado en UML para la integración de subsistemas. Atributos y Subatributos de Calidad de Software. Contratos de Servicios.	<ul style="list-style-type: none"> Diseña una arquitectura basado en UML para integrar subsistemas
	13	25 - 26	Diseño Arquitectónico basado en UML para subsistemas en SOA. Protocolos SOA. Orquestación y Coreografía de servicios web. Diagramas de Componentes para SOA. Diagramas de Despliegue para SOA.	<ul style="list-style-type: none"> Elabora un diagrama de componentes y un diagrama de despliegue para SOA. Utiliza la

			herramienta IBM RSA para elaborar los diagramas
14	27 - 28	Diseño Arquitectónico basado en UML para subsistemas en Arquitecturas para software de dispositivos móviles. Protocolos para software para dispositivos móviles. Diagramas de Componentes para REST. Diagramas de Despliegue para REST.	<ul style="list-style-type: none"> • Elabora un diagrama de componentes y un diagrama de despliegue para arquitecturas REST.
15	29 - 30	Arquitecturas Cloud Computing para software en dispositivos móviles.	<ul style="list-style-type: none"> • Elabora un diagrama de componentes y un diagrama de despliegue para arquitecturas Cloud Computing.
		Evaluación	<ul style="list-style-type: none"> • PRÁCTICA CALIFICADA 3
16	31 - 32	Herramienta IBM RSA.	<ul style="list-style-type: none"> • Utiliza la herramienta IBM RSA para elaborar los diagramas.
17	33 - 34	Generación de Arquitecturas de servicios utilizando Azure SAS.	<ul style="list-style-type: none"> • Gestiona las diversas funcionalidades SAS en un ambiente Azure utilizando máquinas virtuales
18	35	Evaluación	<ul style="list-style-type: none"> • EXAMEN FINAL INDIVIDUAL