UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

VICERRECTORADO ACADÉMICO

FACULTAD DE INGENIERIA DE PRODUCCION Y SERVICIOS

DEPARTAMENTO ACADÉMICO DE INGENIERIA DE SISTEMAS E INFORMATICA

SÍLABO 2023 - A

ASIGNATURA: INGENIERIA DE SOFTWARE I

1. INFORMACIÓN ACADÉMICA

Periodo académico:	2023 - A	
Escuela Profesional:	CIENCIA DE LA COMPUTACIÓN	
Código de la asignatura:	1703132	
Nombre de la asignatura:	INGENIERIA DE SOFTWARE I	
Semestre:	V (quinto)	
Duración:	17 semanas	
Número de horas (Semestral)	**Teóricas:**	2.00
	Prácticas:	0.00
	Seminarios:	0.00
	Laboratorio:	2.00
	Teórico-prácticas:	0.00
Número de créditos:	3	
Prerrequisitos:	CIENCIAS DE LA COMPUTACION II (1702118) BASE DE DATOS I (1702226)	

2. INFORMACIÓN DEL DOCENTE, INSTRUCTOR, COORDINADOR

DOCENTE	GRADO ACADÉMICO	DPTO. ACADÉMICO	HORAS	HORARIO
SARMIENTO CALISAYA, EDGAR	DSc. en Informática	INGENIERIA DE SISTEMAS E INFORMATICA	0	Lun: 08:50-10:30
SARMIENTO CALISAYA, EDGAR	DSc en Informática	INGENIERIA DE SISTEMAS E INFORMATICA	0	Lun: 14:00-15:40

3. INFORMACIÓN ESPECIFICA DEL CURSO (FUNDAMENTACIÓN, JUSTIFICACIÓN)

La tarea de desarrollar software con alta calidad y respetando los costos/cronograma predefinidos, excepto para aplicaciones sumamente simples, exige la ejecución de un proceso de desarrollo bien definido. Los profesionales de esta área deben conocer los diferentes modelos de proceso de desarrollo, para que sean capaces de elegir los métodos, técnicas y herramientas de desarrollo/apoyo más idóneos para cada

proyecto, y ser capaces de trabajar en equipo. Este curso introduce los fundamentos de la ingeniería de software, es decir, las actividades, tareas, prácticas y herramientas relacionadas a las etapas de Ingeniería de Requisitos, Diseño y Construcción de Software.

4. COMPETENCIAS/OBJETIVOS DE LA ASIGNATURA

a) Identificar y documentar los requisitos de software aplicando una técnica de elicitación de requisitos en sesiones de trabajo con partes interesadas, utilizando habilidades de facilitación, como miembro contribuyente de un equipo de requisitos.

b) Especificar los requisitos de software (distinguiendo entre requisitos funcionales y no funcionales) utilizando formatos de especificación estándar y lenguajes que se han seleccionado para el proyecto y ser capaz de describir los requisitos de una manera comprensible para los no expertos, como los usuarios finales, otras partes interesadas o gerentes administrativos, como miembro colaborador de un equipo de requisitos.

c) Verificar y validar los requisitos utilizando técnicas estándar, incluida la inspección, el modelado, la creación de prototipos y el desarrollo de casos de prueba.

d) Presentar a un cliente el diseño de un sistema de software simple usando una notación de modelado (como UML), incluyendo una explicación de cómo el diseño incorporó principios de diseño y estilos o patrones arquitectónicos.

e) Crear documentos de diseño de software que se comuniquen eficazmente con los clientes de diseño de software, como analistas, implementadores, planificadores o mantenedores.

f) Demostrar errores comunes de codificación, construir y depurar programas usando las bibliotecas estándar disponibles con un lenguaje de programación elegido.

g) Aplicar documentación coherente y estándares/estilos/convenciones de programación que contribuyan a la legibilidad, el mantenimiento y la reutilización del software, realizando revisiones de código en componentes de software utilizando una lista de verificación proporcionada.

h) Aplicar principios y enfoques de desarrollo de software que contribuyan a la adaptabilidad , robustez y confiablidad de las soluciones de software complejas.

5. CONTENIDO TEMATICO

PRIMERA UNIDAD

Capítulo I: Ingeniería de Requisitos

- **Tema 01:** Principios de Ingeniería de Software: Proceso de Software, Reglas de Disciplina, Requisito de Software.
- **Tema 02:** Los Límites del Sistema y el Contexto.
- **Tema 03:** Elicitación de Requisitos
- **Tema 04:** Especificación de Requisitos: Lenguaje Natural (Casos de Uso, Historias de Usuario, BDD) y Modelos (NFR framework)
- **Tema 05:** Análisis de Requisitos (Verificación y Validación).
- **Tema 06:** Gestión de Requisitos, Documento de Especificación de Requisitos y Herramientas.

SEGUNDA UNIDAD

Capítulo II: Diseño de Software

- **Tema 07:** 7. Principios de diseño de software

Tema 08: 8. Paradigmas de diseño : estructurado, orientado a objetos, aspectos, funciones, servicios

Tema 09: 9. Modelos estructurales y de comportamiento del diseño orientado a objetos

Tema 10: 10. Arquitectura de software: Fundamentos y Documentación

Tema 11: 11. Estilos y patrones de arquitectura: filtros y tuberías, capas, eventos, micro kernel, plugins, micro servicios

Tema 12: 12. Componentes de Software, Bibliotecas, Frameworks y Middlewares.

TERCERA UNIDAD

Capítulo III: Construcción de Software

Tema 13: Estilos de programación y Programación Defensiva

Tema 14: Mecanismos para construcción de programas de calidad: Normas/Estándares/Estilos/Guías/Convenciones de codificación, Codificación Legible (clean code).

Tema 15: Estrategias de Integración: top-down, bottom-up, sandwich

Tema 16: Prácticas de desarrollo de software orientado a objetos: Principios SOLID

Tema 17: Enfoques de desarrollo de software: Domain-driven Design ? DDD, Clean Architecture

6. PROGRAMACIÓN DE ACTIVIDADES DE INVESTIG. FORMATIVA Y RESPONSABILIDAD SOCIAL

6.1. Métodos

Método expositivo en las clases teóricas,
Método de elaboración conjunta en las aulas prácticas del proyecto de desarrollo

6.2. Medios

Pizarra acrílica, plumones, cañón multimedia, videos, software y guías de práctica

6.3. Formas de organización

i. Clases teóricas: Conceptos
ii. Practicas: Aplicación de Conceptos
iii. Laboratorio: Uso de Recursos de Software
iv. Otro: Presentaciones, videos

6.4. Programación de actividades de investigación formativa y responsabilidad social

i. Investigación Formativa: Proyecto Final de Investigación o Implementación
ii. Responsabilidad Social: Difusión del Pensamiento Computacional en la Región Arequipa

7. CRONOGRAMA ACADÉMICO

SEMANA	TEMA	DOCENTE	%	ACUM.
1	Principios de Ingeniería de Software: Proceso de Software, Reglas de Disciplina, Requisito de Software.	E. Sarmiento	6	6.00
2	Los Límites del Sistema y el Contexto.	E. Sarmiento	4	10.00
2	Elicitación de Requisitos	E. Sarmiento	6	16.00
3	Especificación de Requisitos: Lenguaje Natural (Casos de Uso, Historias de Usuario, BDD) y Modelos (NFR framework)	E. Sarmiento	6	22.00

4	Análisis de Requisitos (Verificación y Validación).	E. Sarmiento	6	28.00
5	Gestión de Requisitos, Documento de Especificación de Requisitos y Herramientas.	E. Sarmiento	6	34.00
7	7. Principios de diseño de software	E. Sarmiento	6	40.00
7	8. Paradigmas de diseño : estructurado, orientado a objetos, aspectos, funciones, servicios	E. Sarmiento	6	46.00
8	9. Modelos estructurales y de comportamiento del diseño orientado a objetos	E. Sarmiento	6	52.00
9	10. Arquitectura de software: Fundamentos y Documentación	E. Sarmiento	6	58.00
10	11. Estilos y patrones de arquitectura: filtros y tuberías, capas, eventos, micro kernel, plugins, micro servicios	E. Sarmiento	6	64.00
11	12. Componentes de Software, Bibliotecas, Frameworks y Middlewares.	E. Sarmiento	4	68.00
13	Estilos de programación y Programación Defensiva	E. Sarmiento	6	74.00
14	Mecanismos para construcción de programas de calidad: Normas/Estándares/Estilos/Guías/Convenciones de codificación, Codificación Legible (clean code).	E. Sarmiento	6	80.00
15	Estrategias de Integración: top-down, bottom-up, sandwich	E. Sarmiento	6	86.00
15	Prácticas de desarrollo de software orientado a objetos: Principios SOLID	E. Sarmiento	6	92.00
16	Enfoques de desarrollo de software: Domain-driven Design ? DDD, Clean Architecture	E. Sarmiento	8	100.00

8. ESTRATEGIAS DE EVALUACIÓN

8.1. Evaluación del aprendizaje

1.- Evaluación Continua. Se evaluará durante todo el semestre a los estudiantes considerando:
1.1. Su actitud solidaria o egoísta, su interés por aprender, el que sea autodidacta y aplicación de los contenidos, participación en clase, el trabajo de investigación formativa, participación en prácticas de laboratorio, tanto para el primer parcial (EC1), segundo parcial (EC2) y tercer parcial (EC3)

2.- Evaluación Periódica.
2.1 Primer Examen (EP1)
2.2 Segundo Examen (EP2)
2.3 Proyecto Final (PF) de Investigación o Implementación

3.- Examen Subsanación o Recuperación (Sustitutorio):

8.2. Cronograma de evaluación

EVALUACIÓN	FECHA DE EVALUACIÓN	EXAMEN TEORÍA	EVAL. CONTINUA	TOTAL (%)
Primera Evaluación Parcial	29-05-2023	20%	9%	**29%**
Segunda Evaluación Parcial	03-07-2023	20%	9%	**29%**
Tercera Evaluación Parcial	27-07-2023	27%	15%	**42%**
			TOTAL	**100%**

9. REQUISITOS DE APROBACIÓN DE LA ASIGNATURA

Se tomará en cuenta para la aprobación del estudiante, las normas establecidas en el Reglamento General de Evaluación del proceso enseñanza aprendizaje de la UNSA:

a) El alumno tendrá derecho a observar o en su defecto a ratificar las notas consignadas en sus evaluaciones, después de ser entregadas las mismas por parte del profesor, salvo el vencimiento de plazos para culminación del semestre académico, luego del mismo, no se admitirán reclamaciones, alumno que no se haga presente en el día establecido, perderá su derecho a reclamo.

b) Para aprobar el curso el alumno debe obtener una nota igual o superior a 10.5, en el promedio final.

c) El redondeo, sólo se efectuará en el cálculo del promedio final, quedado expresó, que las notas parciales, no se redondearán individualmente.

d) El alumno que no tenga alguna de sus evaluaciones y no haya solicitado evaluación de rezagados en el plazo oportuno, se le considerará como abandono.

e) Los casos particulares por los cuales el alumno no pudo cumplir con su evaluación en el tiempo establecido, podrá tramitar ante la dirección de escuela, su respectiva justificación, con la cual, el profesor tendrá la obligación de tomarle una nueva evaluación, la misma que sustituirá, la nota en cuestión.

f) El estudiante quedará en situación de ?abandono? si el porcentaje de asistencia es menor al ochenta (80%) por ciento en las actividades que requieran evaluación continua (Prácticas, talleres, seminarios, etc.).

g) A continuación, se muestra la fórmula de Promedio Final (PF):

PF= EC1*0.09 + EP1*0.20 + EC2*0.09 + EP2*0.20 + EC3*0.15 + PF*0.27

10. BIBLIOGRAFIA: AUTOR, TÍTULO, AÑO, EDITORIAL

10.1. Bibliografía básica obligatoria

a) Pressman, R. S. and Maxim, B. (2014). Ingeniería de Software: Un Enfoque Práctico. McGraw-Hill, 8th edition.

b) Sommerville, I. (2010). Ingeniería de Software. Addison-Wesley, 9th edition.

c) McConnell S. (2011). Code Complete 2: A practical Handbook of software construction. Segunda edición. Microsoft Press.

d) Winters, T., Manshreck, T., & Wright, H. 2020. Software engineering at google: Lessons learned from programming over time. O'Reilly Media.

10.2. Bibliografía de consulta

a) Karl E Wiegers, Joy Beatty. 2003. Software Requirements.

b) Klaus Pohl, Chris Rupp. 2015. Requirements Engineering Fundamentals, Rocky Nook Inc.

c) M Richards. 2015. Software architecture patterns.
https://www.oreilly.com/programming/free/software-architecture-patterns.csp

d) Martin Fowler, Dave Rice, Matthew Foemmel, Edward Hieatt, Robert Mee, and Randy Stafford. 2002. Patterns of Enterprise Application Architecture

e) Robert C. Martin. 2017. Clean Architecture: A Craftsman?s Guide to Software Structure and Design

f) Robert C. Martin. 2008. Clean Code: A Handbook of Agile Software Craftsmanship

g) Lopes, Cristina Videira. Exercises in programming style. CRC Press, 2014. Disponible en:
http://ecoop14.it.uu.se/programme/Lopes_ECOOP_2014.pdf
https://github.com/crista/exercises-in-programming-style

h) Eric Evans. Domain-Driven Design: Tackling Complexity in the Heart of Software. 2003

i) Henderson, F. 2017. Software engineering at Google. arXiv preprint arXiv:1702.01715.

Arequipa, 30 de Mayo del 2023

SARMIENTO CALISAYA, EDGAR