SILABO

FACULTAD INGENIERIA ESTADISTICA E INFORMATICA
ESCUELA PROFESIONAL INGENIERIA ESTADISTICA E INFORMATICA
PROGRAMA DE ESTUDIOS CARRERA PURA

I. INFORMACIÓN GENERAL

1.1 Identificación Académica

a)	Curso	**INGENIERIA DE SOFTWARE I**
b)	Código	EST319
c)	Prerequisito	EST310 - LENGUAJES DE PROGRAMACION III, EST315 - GESTION DE PROYECTOS TICS
d)	Número de Horas	03h teóricas, 02h prácticas, Total 05 horas
e)	Créditos	04
f)	Número de Horas virtuales	90
g)	Año y Semestre Académico	2025-II
h)	Ciclo de Estudios	VII
i)	Duración	Del 25 de Agosto al 19 de Diciembre del 2025 (17 semanas)
j)	Área Curricular	Estudios de especialidad
k)	Características del Curso	Investigación, desarrollo e Innovación

1.2 Docente

a)	Apellidos y Nombres	LAURA MURILLO RAMIRO PEDRO
b)	Condición y Categoria	Docente Contratado
c)	Especialidad	Doctor en Ciencias de la Computación, Ingeniero Estadístico e Informático

a)	Apellidos y Nombres	LOPEZ CUEVA MILTON ANTONIO
b)	Condición y Categoria	NOMBRADO
c)	Especialidad	Doctor en Estadística e Informática, Ingeniero Estadístico e Informático

1.3 Ambiente donde se realizó el aprendizaje

a) Salon de clases, aula virtual, laboratorio de computo

II. SUMILLA

El curso de Ingeniería de Software II profundiza en los principios avanzados de desarrollo de software, centrándose en metodologías ágiles, diseño de arquitecturas escalables, pruebas automatizadas, gestión de calidad y despliegue de software en entornos reales. Se abordarán frameworks modernos, patrones de diseño, DevOps y métricas de calidad para formar profesionales capaces de liderar proyectos de software robustos y mantenibles

III. PERFIL DEL EGRESADO EN RELACIÓN AL CURSO

Al finalizar el curso, el estudiante será capaz de:

- Diseñar arquitecturas de software escalables y mantenibles.
- Aplicar metodologías ágiles (Scrum, Kanban) en proyectos reales.
- Implementar pruebas automatizadas (unitarias, de integración y E2E).
- Utilizar herramientas de CI/CD (GitHub Actions, Jenkins, Docker).
- Gestionar la calidad del software mediante métricas y estándares (ISO 25010).

IV. LOGRO DE APRENDIZAJE DEL CURSO

Al finalizar el curso, el estudiante:

- Analiza requerimientos complejos y propone soluciones técnicas escalables.
- Diseña arquitecturas de software usando patrones y buenas prácticas.
- Implementa pipelines de CI/CD para automatizar despliegues.
- Evalúa la calidad del software mediante métricas y estándares.

V. TRATAMIENTO DE UNIDADES DIDÁCTICAS

UNIDAD 1	UNIDAD 1
LOGROS DE APRENDIZAJE DE LA UNIDAD Explica adecuadamente las metodologías de desarrollo	
TIEMPO DE DESARROLLO	Del 25 de Agosto al 27 de Octubre del 2025 (Total 45 horas)
HORAS DE ENSEÑANZA VIRTUAL/UNIDAD	45

SEMANAS	CRITERIOS DE DESEMPEÑO	CONOCIMIENTOS
Semana 1	Demuestra comprensión de los principios ágiles mediante lacomparación estructurada con modelos tradicionales	Historia y evolución de metodologías de desarrollo, ágil

Semana 2	Aplica Scrum correctamente en la planificación de un sprint simulando roles y artefactos	Roles Scrum (SM/PO/Team), Artefactos (Backlogs),Ceremonias (Daily/Review)
Semana 3	Diseña arquitecturas limpias aplicando el principio de inversión de dependencias	MVC, Clean Architecture, Principio SOLID de inversión dedependencias
Semana 4	Implementa un microservicio funcional con comunicación inter-servicios	API Gateway, Service Discovery, Comunicaciónsíncrona/asíncrona
Semana 5	Modela flujos transaccionales complejos usando patrones Event-Driven	Event Sourcing, CQRS, Patrón Saga para transaccionesdistribuidas
Semana 6	Refactoriza código aplicando patrones creacionales correctamente	Factory Method, Singleton thread-safe, Builder Pattern
Semana 7	Integra sistemas legacy mediante adaptadores y compuestos estructurales	Adapter Pattern, Composite Pattern, Proxy dinámico
Semana 8	Implementa estrategias de comportamiento para algoritmos complejos\	Observer, Strategy, State Pattern
Semana 9	Desarrolla pruebas unitarias que cubran >80% del código (métricas verificables)	JUnit/Mockito, Test Coverage (JaCoCo), TDD/BDD
PORCENTAJE DE AVANCE ACADÉMICO DE LA UNIDAD: 50%		

* El docente deberá programar “actividades de retroalimentación” los cuales se desarrollarán a lo largo de cada unidad didáctica con el fin de asegurar los adecuados aprendizajes de los estudiantes.

UNIDAD 2	UNIDAD 2	
LOGROS DE APRENDIZAJE DE LA UNIDAD Explica adecuadamente los estándares de seguridad en el desarrollo de software.		
TIEMPO DE DESARROLLO	Del 27 de Octubre al 19 de Diciembre del 2025 (Total 40 horas)	
HORAS DE ENSEÑANZA VIRTUAL/UNIDAD	45	
SEMANAS	**CRITERIOS DE DESEMPEÑO**	**CONOCIMIENTOS**
Semana 10	Configura pipelines CI/CD funcionales con ejecución automatizada	GitHub Actions/Jenkins, Docker, Orchestration básica
Semana 11	Evalúa calidad de código mediante métricas estándar en herramientas SonarQube	ISO 25010, Métricas de código (deuda técnica, code smells)
Semana 12	Aplica principios DevSecOps en despliegues automatizados	OWASP Top 10, SAST/DAST, Administración de contenedores
Semana 13	Diseña arquitecturas seguras con autenticación/autorización robusta	JWT, OAuth2.0, RBAC/ABAC, Zero-Trust Architecture
Semana 14	Diagnostica problemas en legacy code mediante técnicas de refactoring	Code Smells, Refactoring Patterns (Extract Method, etc.)
Semana 15	Propone estrategias de modernización para sistemas legacy	Strangler Pattern, Branch by Abstraction
Semana 16	Documenta arquitecturas técnicas usando estándares C4 y UML 2.5	Diagramas C4 (Contexto/Contenedores/Componentes/Código)
Semana 17	Presenta proyectos cumpliendo estándares ISO/IEC 26515	Documentación de arquitectura, Manuales técnicos
PORCENTAJE DE AVANCE ACADÉMICO DE LA UNIDAD: 100%		

* El docente deberá programar “actividades de retroalimentación” los cuales se desarrollarán a lo largo de cada unidad didáctica con el fin de asegurar los adecuados aprendizajes de los estudiantes.

VI. ESTRATEGIAS METODOLÓGICAS

6.1 De Enseñanza

- Se hará uso de ejemplos teóricos y prácticos en las sesiones de clases y laboratorios.
- Se utilizará textos universitarios, tanto del docente como el de la biblioteca.
- Se propondrá seminarios y talleres para el análisis crítico de la teoría y la práctica
- Se registrará la teoría y la práctica en forma constante, así como también las asistencias.

6.2 De Aprendizaje

Control de prácticas, trabajos teóricos y prácticos y conocimientos a través de laboratorio de computadoras y cuestionario de preguntas con dinámicas grupales y foros en las sesiones de clases. Como también se agruparán para realizar trabajos de investigación, proyección social y extensión universitaria para propiciar la cultura académica e investigativa

6.3 De Investigación Formativa

Búsqueda de información complementaria en textos, revistas e internet.

6.4 De Responsabilidad Social Universitaria

Participación en guías de prácticas saludables, participación y colaboración de trabajos grupales en el cuidado del medio ambiente.

6.5 De Enseñanza Virtual

Uso de páginas web y redes sociales en el proceso de búsqueda y análisis crítico de conceptos teóricos del componente curricular.

VII. MEDIOS Y MATERIALES DIDÁCTICOS

Auditivo

- En forma directa con la voz del docente
- Permanente comunicación docente – estudiante- Equipo de sonido del laboratorio

Visuales

- Separatas, guías de estudio.
- Textos universitarios.
- Hojas de trabajo individual y de grupo
- pizarra, multimedios e internet.

VIII. EVALUACIÓN DEL APRENDIZAJE

8.1 Logro de aprendizaje, evidencias de desempeño, ponderación, técnicas e instrumentos de evaluación.

UNIDAD	LOGROS DE APRENDIZAJE	EVIDENCIAS DESEMPEÑO: De acción, objeto o producto (%)	PONDERACIÓN (Obligatorio en base a 100%)	TÉCNICAS	INSTRUMENTOS
1	Explica adecuadamente las metodologías de desarrollo	Desarrolla los conceptos de las metodologías de desarrollo ágil.	50%	- Exposición - Examen	- Ejemplos - Bibliografía
2	Explica adecuadamente los estándares de seguridad en el desarrollo de software.	Desarrolla los conceptos de seguridad con DevSecOps y los traza en una aplicación real.	50%	- Exposición - Examen	- Ejemplos - Bibliografía

8.2 Evidencias de aprendizaje del semestre académico.

LOGRO DE APRENDIZAJE DEL CURSO	EVIDENCIAS: ACCIÓN, OBJETO o PRODUCTO	FECHA DE PRESENTACIÓN
Al finalizar el curso, el estudiante: • Analiza requerimientos complejos y propone soluciones técnicas escalables. • Diseña arquitecturas de software usando patrones y buenas prácticas. • Implementa pipelines de CI/CD para automatizar despliegues. • Evalúa la calidad del software mediante métricas y estándares.	Última semana del semestre académico	Logra comprender y asimilar los conceptos

• Evidencias que deben ser socializados con la comunidad como parte de la responsabilidad social universitaria. Por la emergencia sanitaria podría optarse por la difusión vía página web y redes sociales.

8.3 Calificación

La fórmula para la obtención del promedio final del curso es la siguiente:

Promedio Final = (50%)IUPP+(50%)IIUPP

Donde:
IUPP : Primero unidad promedio parcial
IIUPP : Segundo unidad promedio parcial

IX. FUENTES DE INFORMACIÓN

9.1 Bibliográficas

Básica

1. Pressman, R. S., & Maxim, B. R. (2020).
Software Engineering: A Practitioner's Approach (9th ed.). McGraw-Hill.
ISBN: 978-1260548009
Cubre el ciclo de vida del software, modelos de proceso y gestión de proyectos

2. Sommerville, I. (2016).
Software Engineering (10th ed.). Pearson.
ISBN: 978-0133943030
Aborda requisitos, arquitectura, pruebas y evolución de software.

3. Martin, R. C. (2009).
Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall.
ISBN: 978-0132350884
Best practices para escribir código mantenible y legible.

Complementarias

Electrónicas

IEEE (2021).
IEEE Std 730-2021: Standard for Software Quality Assurance.
Normas internacionales para calidad de software.

ISO/IEC 25010 (2011).
Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE).
Modelo de calidad de software ampliamente adoptado.

Producción intelectual del docente relacionado con el curso

Puno, Diciembre del 2025