



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
VICEDECANATO DE INVESTIGACIÓN Y POSGRADO
UNIDAD DE POSGRADO

SILABO

1. INFORMACIÓN GENERAL

1.1	Nombre de la asignatura	:	ENTORNOS METODOLÓGICOS DE DESARROLLO DE SOFTWARE
1.2	Código de la asignatura	:	S2025109
1.3	Tipo de Asignatura	:	Teoría-Práctica
1.4	Número de Semanas	:	16
1.5	Créditos	:	4.0
1.6	Horas semanales	:	3
1.7	Número de horas	:	48 horas
1.8	Semestre Académico	:	2025-II
1.9	Pre-requisito	:	Ninguno
1.10	Docente	:	Víctor Hugo Bustamante Olivera
1.11	Correo electrónico	:	vbustamanteo@unmsm.edu.pe

2. SUMILLA:

Asignatura que corresponde al periodo Investigación, es de naturaleza Teórica-práctica y de modalidad semipresencial. Tiene por propósito conocer el enfoque ágil de gestión, saber cómo utilizar las prácticas ágiles de gestión de proyectos de software y de equipos de personas. La gestión ágil de proyectos es el primer paso natural antes de la gestión ágil de servicios, que también actualmente se propone llevar a cabo desde un punto de vista ágil. Manejo de Metodologías ágiles: Scrum, XP, Kanban y otras. Al finalizar el estudiante deberá presentar un informe aplicando los tópicos tratados alineados a una línea de investigación.

3. ASPECTOS DEL PERFIL PROFESIONAL

El curso contribuye al logro de las siguientes competencias del perfil del egresado de la maestría:

- 3.1 Domina las áreas fundamentales de la gestión de proyecto de software
- 3.2 Lidera equipos de investigación y desarrollo vinculado al desarrollo de sistemas de información.
- 3.3 Formula políticas y estrategias para el desarrollo de la industria del software
- 3.4 Gerencia y desarrolla proyectos de software alineados a los estándares internacionales de calidad.
- 3.5 Concilia objetivos en conflicto de un proyecto encontrando compromisos aceptables dentro de las limitaciones de costo, tiempo, conocimiento, sistemas existentes y organizacionales.
- 3.6 Apoya las actividades científicas multidisciplinarias que requieran de la ingeniería de sistemas y computación.
- 3.7 Práctica un comportamiento ético y profesional.
- 3.8 Ejerce la docencia a alto nivel en las universidades.

4. COMPETENCIAS DE LA ASIGNATURA

Las competencias que los estudiantes van lograr en el curso son los siguientes:

- 4.1 Enjuicia la práctica actual que existe en gestión de proyectos de software
- 4.2 Analiza e identifica los procesos que hay en gestión de proyectos de software
- 4.3 Ordena y organiza las entregas parciales del software
- 4.4 Planifica y gestiona la gestión de proyectos de software
- 4.5 Registra, sintetiza y asegura el cierre del proyecto de software

5. LOGROS DE APRENDIZAJE

Al finalizar el curso, el estudiante habrá logrado los siguientes resultados de aprendizaje, que se distribuyen a lo largo de las cuatro unidades didácticas:

A nivel Cognitivo (Saber y Entender):

1. Comprende y distingue los principios, valores y prácticas de las principales metodologías



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
VICEDECANATO DE INVESTIGACIÓN Y POSGRADO

UNIDAD DE POSGRADO

ágiles (Scrum, Kanban, XP) y los compara con los enfoques tradicionales.

2. **Analiza** cómo la automatización a través de herramientas de Integración Continua y Entrega Continua (CI/CD) mejora el ciclo de vida del software.
3. **Evaluá** la idoneidad de diferentes metodologías y herramientas para distintos tipos de proyectos y contextos organizacionales.
4. **Identifica y relaciona** el rol de la cultura DevOps con la agilidad y la eficiencia en la entrega de productos de software.
5. **Reconoce** el valor de las arquitecturas de microservicios y el uso de contenedores (Docker) en entornos de desarrollo modernos.
6. **Examina** el rol de la gestión de riesgos y la calidad en el contexto de un proyecto ágil.

A nivel de Habilidades (Saber Hacer):

1. **Aplica** los roles, artefactos y eventos de Scrum en un proyecto de software simulado.
2. **Utiliza** herramientas de colaboración y control de versiones (Git) para gestionar el código fuente de forma efectiva en un entorno de equipo.
3. **Desarrolla** software de mayor calidad utilizando prácticas como el Desarrollo Dirigido por Pruebas (TDD) y el Desarrollo Dirigido por Comportamiento (BDD).
4. **Formula** requisitos de software de forma ágil, empleando Historias de Usuario.
5. **Elabora** un informe de investigación que aplique los conceptos del curso a una problemática real, demostrando la conexión entre la teoría y la práctica.
6. **Sustenta** y defiende de manera coherente las decisiones metodológicas y los resultados de su proyecto de investigación

6. PROGRAMACIÓN DE CONTENIDOS

UNIDAD 1: Fundamentos y Metodologías Tradicionales

En esta unidad, se establecen las bases teóricas de la gestión de proyectos de software. Se explora la evolución desde los modelos clásicos hasta el surgimiento de las metodologías ágiles, entendiendo el contexto y las motivaciones de este cambio de paradigma.

Logros de Aprendizaje de la Unidad:

- Comprende los fundamentos del desarrollo de software y el impacto de las metodologías en el éxito de los proyectos.
- Distingue las características, ventajas y desventajas de los modelos tradicionales (Cascada, Espiral).
- Analiza los principios y valores del Manifiesto Ágil y reconoce su importancia en la industria actual.
- Identifica los roles, artefactos y eventos de Scrum y Kanban como los marcos de trabajo más populares.

Sem.	Contenido	Actividades	Recursos	Estrategias
1	Fundamentos del Desarrollo de Software y Modelos Tradicionales	- Discusión de experiencias de proyectos fallidos. - Análisis comparativo de modelos.	- Diapositivas. - Casos de estudio. - Pizarra interactiva.	- Exposición magistral - Discusión de casos de fracaso de proyectos. - Análisis comparativo de modelos tradicionales.
2	Introducción a Metodologías Ágiles	- Debate sobre el Manifiesto Ágil. - Identificación de diferencias clave con modelos tradicionales.	- Video del Manifiesto Ágil. - Texto del Manifiesto. - Artículos sobre el tema.	Sesión interactiva, análisis crítico, discusión guiada.
3	Scrum	- Taller de asignación de roles - Creación de un Product Backlog. - Demostración de un Sprint Planning.	- Pizarra interactiva. - Plantillas de Scrum. - Herramienta digital (Jira, Trello).	Aprendizaje experiencial, taller práctico, simulación.
4	Kanban	- Taller de creación de un	- Pizarra.	Aprendizaje práctico, modelado de



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
VICEDECANATO DE INVESTIGACIÓN Y POSGRADO

UNIDAD DE POSGRADO

		tablero Kanban - Medición del flujo (Lead Time, Cycle Time).	- Software de gestión de proyectos.	procesos, análisis de métricas.
--	--	---	-------------------------------------	---------------------------------

UNIDAD 2: Profundización en Marcos de Trabajo y Prácticas Ágiles

Esta unidad profundiza en la aplicación práctica de los marcos ágiles y las técnicas asociadas para gestionar equipos y proyectos de manera efectiva. Se enfoca en las habilidades necesarias para una correcta ejecución.

Logros de Aprendizaje de la Unidad:

- Aplica técnicas de gestión ágil como la creación de Historias de Usuario y la estimación con Planning Poker.
- Diferencia y describe las prácticas centrales de la Programación Extrema (XP).
- Utiliza herramientas de colaboración y control de versiones (Git) para gestionar el código fuente de forma eficiente.
- Defiende y argumenta las decisiones metodológicas tomadas en el proyecto de investigación parcial.

Sem.	Contenido	Actividades	Recursos	Estrategias
5	Programación Extrema (XP)	- Discusión de los valores de XP - Demostración de TDD y programación en pares.	- Videos - Ejemplos de código. - Pizarra.	Clase magistral, demostración en vivo, análisis de casos.
6	Historias de Usuario	- Taller de escritura de Historias de Usuario para un proyecto. - Identificación de criterios de aceptación.	- Plantillas de historias. - Ejemplos de proyectos.	Aprendizaje práctico, trabajo en grupo, feedback entre pares.
7	Estimación en Entornos Ágiles	- Taller de Planning Poker - Cálculo de la velocidad del equipo.	- Cartas de Planning Poker. - Hojas de cálculo. - Herramienta digital de estimación.	Taller lúdico, ejercicio de cálculo, discusión de resultados.
8	Examen parcial	- Examen teórico/práctico. - Entrega del avance del proyecto de investigación.	- Plataforma virtual de exámenes - Rúbrica de evaluación.	Evaluación individual, entrega de proyecto.

UNIDAD 3: Herramientas y Entornos de Automatización

En esta unidad, el enfoque se desplaza de la teoría a la práctica técnica. Se explora cómo las herramientas y la automatización son pilares de la agilidad moderna, cubriendo conceptos como la integración, la entrega y la cultura DevOps.

Logros de Aprendizaje de la Unidad:

- Desarrolla software de mayor calidad utilizando el Desarrollo Dirigido por Pruebas (TDD).
- Diseña y configura un pipeline básico de Integración Continua (CI) y Entrega Continua (CD).
- Identifica las prácticas y herramientas clave de la cultura DevOps.
- Evalúa la aplicación de arquitecturas de microservicios y el uso de contenedores (Docker) para el despliegue de aplicaciones.

Sem.	Contenido	Actividades	Recursos	Estrategias
9	TDD	- Taller de programación aplicando el ciclo Red-Green-Refactor.	- Entorno de desarrollo (IDE). - Lenguaje de programación. - Framework de pruebas unitarias.	Práctica guiada, programación en pares, retroalimentación instantánea.



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
VICEDECANATO DE INVESTIGACIÓN Y POSGRADO

UNIDAD DE POSGRADO

10	CI/CD	- Configuración de un pipeline básico en GitHub Actions.	- Cuenta de GitHub. - Repositorio de código. - Servidor de integración.	Taller guiado, trabajo individual, demostración.
11	DevOps	- Debate sobre la cultura DevOps - Análisis de casos de estudio (Amazon, Netflix).	- Videos. - Artículos. - Casos de estudio. - Pizarra interactiva.	Análisis crítico, debate en grupo, exposición magistral.
12	Contenedores (Docker)	- Taller de creación y despliegue de un contenedor Docker.	- Computadora con Docker instalado. - Archivos de configuración.	Taller práctico, demostración en vivo, aprendizaje guiado.

UNIDAD 4: Estrategias de Implementación y Líneas de Investigación

La última unidad conecta el conocimiento adquirido con el contexto de la investigación y la implementación en el mundo real. Se exploran las tendencias avanzadas y se culmina con la aplicación de lo aprendido en un informe de investigación final.

Logros de Aprendizaje de la Unidad:

- Analiza y propone estrategias para la adopción de metodologías ágiles en grandes organizaciones.
- Reconoce el papel de la gestión ágil de servicios y otras tendencias emergentes en la industria.
- Elabora un informe de investigación aplicando los conceptos del curso a una línea de investigación.
- Argumenta y sustenta de manera coherente los resultados de la investigación.

Sem.	Contenido	Actividades	Recursos	Estrategias
13	Agile a Escala	- Presentación de frameworks de escalado (SAFe, LeSS). - Discusión de los retos de la implementación a escala.	- Diapositivas. - Casos de estudio.	Clase magistral, análisis de casos, debate en grupo.
14	Tendencias en Metodologías	- Exposición de estudiantes sobre una tendencia (AIOps, Low-Code).	- Videos - Artículos - Pizarra.	Investigación, exposición, debate.
15	Presentación de Proyectos	- Exposición y defensa de los informes de investigación.	- Computadora y proyector. - Informes escritos. - Rúbrica de evaluación.	Presentación oral, defensa de proyecto, evaluación por pares.
16	EXAMEN FINAL	- Examen teórico/práctico. - Entrega del informe final de investigación.	- Plataforma virtual de exámenes. - Rúbrica de evaluación.	Evaluación individual, entrega de proyecto.

7. ESTRATEGIA DIDÁCTICA

El curso se desarrollará bajo un enfoque teórico-práctico, combinando métodos activos y herramientas tecnológicas para garantizar el logro de competencias. Se organizará de la siguiente manera:

A. Métodos de Enseñanza-Aprendizaje



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
VICEDECANATO DE INVESTIGACIÓN Y POSGRADO
UNIDAD DE POSGRADO

- Clases expositivas
 - Aprendizaje basado en problemas
 - Discusión guiada
 - Talleres colaborativos
 - Estudio de casos
- B. Técnicas de Evaluación Formativa
- Evaluaciones sorpresa: Preguntas cortas al inicio de clases para reforzar conceptos previos.
 - Rúbricas de laboratorio: Criterios claros para evaluar informes (ej.: completitud, originalidad, uso de herramientas).
 - Peer review (evaluación entre pares): Revisión cruzada de trabajos.
- C. Recursos Didácticos
- Visuales:
 - Diagramas.
 - Videos explicativos .
 - Interactivos:
 - Herramientas en línea.
 - Bibliográficos:
 - Capítulos clave del libro + artículos complementarios.
- D. Adaptaciones para Diferentes Estilos de Aprendizaje
- Aprendices visuales: Diagramas animados y simulaciones.
 - Aprendices kinestésicos: Talleres de construcción manual.
 - Aprendices teóricos: Discusión de temas presentados.

8. EVALUACIÓN

La evaluación de los participantes es integral, continua y permanente. Se tomarán en cuenta los conocimientos adquiridos y la capacidad de los participantes, de acuerdo con los siguientes criterios:

- Responsabilidad del participante respecto a la asistencia, la cuál será registrada a través del SUM, el cual tendrá una tolerancia de 15 minutos.
- Se tendrá en cuenta la intervención activa de los participantes durante las clases, los estudios de casos, talleres evaluativos y presentación de proyectos.

El sistema de evaluación para aprobar la asignatura exige a los alumnos una nota mínima de 13, resultante de las 3 notas indicadas por los siguientes conceptos:

Notas	Instrumento de Evaluación	Porcentaje	Descripción
EP	Examen Parcial	30 %	Examen Parcial
Ev.C	Nota de evaluación continua	40 %	Evaluación continua
EF	Examen Final	30 %	Examen Final
Fórmula			Promedio final = 0.3 EF + 0.4 Ev.C + 0.3 EP

En la evaluación se considerará los siguientes parámetros:

- En las fechas de evaluaciones, la tolerancia máxima de ingreso es de 15 minutos, después de esto se considerará ausente.
- En caso de dolo en cualquier evaluación, esta se sancionará con la nota 0 (cero).

9. BIBLIOGRAFÍA

Libros de texto

- Beck, K. (2000). Extreme Programming Explained: Embrace Change. Addison-Wesley.
- Cohn, M. (2006). Succeeding with Agile: Software Development Using Scrum. Addison-Wesley.
- Schwaber, K., & Sutherland, J. (2020). La Guía Definitiva de Scrum: Las Reglas del Juego.



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
VICEDECANATO DE INVESTIGACIÓN Y POSGRADO
UNIDAD DE POSGRADO

Scrum.org.

- Pressman, R. S., & Maxim, B. R. (2020). Ingeniería del Software: Un Enfoque Práctico (9a ed.). McGraw-Hill.

Artículos de Lectura

- Bäcklander, G. (2019). Doing complexity leadership theory: How agile coaches at Spotify practise enabling leadership. *Creativity and innovation management*, 28(1), 42-60.
- Sohaib, O., Solanki, H., Dhaliwa, N., Hussain, W., & Asif, M. (2019). Integrating design thinking into extreme programming. *Journal of Ambient Intelligence and Humanized Computing*, 10, 2485-2492.
- Desai, R., & Nisha, T. N. (2021, July). Best practices for ensuring security in devops: A case study approach. In *Journal of Physics: Conference Series* (Vol. 1964, No. 4, p. 042045). IOP Publishing.
- Battina, D. S. (2017). Best practices for ensuring security in Devops: A case study approach. *International Journal of Innovations in Engineering Research and Technology*, 4(11), 38-45.
- Wazir, S., Kashyap, G. S., & Saxena, P. (2023). Mlops: A review. *arXiv preprint arXiv:2308.10908*.
- Testi, M., Ballabio, M., Frontoni, E., Iannello, G., Moccia, S., Soda, P., & Vessio, G. (2022). MLOps: a taxonomy and a methodology. *IEEE Access*, 10, 63606-63618.