



GUÍA DOCENTE 2026-2027
Metodología ágiles y diseño de sistemas software

1. Denominación de la asignatura:

METODOLOGÍAS ÁGILES Y DISEÑO DE SISTEMAS SOFTWARE

Titulación

Grado en Tecnologías Digitales para la Empresa

Código

8999

2. Materia o módulo a la que pertenece la asignatura:

Ingeniería del Software

3. Departamento(s) responsable(s) de la asignatura:

Digitalización

4.a Profesor/a que imparte la docencia (Si fuese impartida por más de uno/a incluir todos/as) :

Mohamed Saban

4.b Coordinador/a de la asignatura

Carlos Cambra

5. Curso y semestre en el que se imparte la asignatura:

Curso 3º - Semestre 6º

6. Tipo de la asignatura: (Básica, obligatoria u optativa)

Obligatoria



7. Requisitos de formación previos para cursar la asignatura:

Ninguno

8. Número de créditos ECTS de la asignatura:

6

9. Competencias // Resultados de aprendizaje

CT1 - Capacidad de análisis y síntesis.
CT3 - Comunicación oral y escrita en la lengua nativa.
CT5 - Capacidad de gestión de la información.
CT7 - Trabajo en un equipo de carácter multidisciplinar
CT8 - Aprendizaje y trabajo autónomo. Formación continua y permanente.
CT9 - Adaptación a nuevas situaciones.
CT10 - Motivación por la calidad.
CT11 - Comunicarse con personas expertas y no expertas en la materia.
CT12 - Elaborar y defender argumentos.
CT13 - Creatividad.
CT17 - Trabajo en equipo.
CT18 - Trabajo en un contexto internacional.
CT19 - Adaptación a nuevas situaciones.
GC1.25 Conocer y comprender los diferentes tipos de sistemas (tiempo compartido, tiempo real, etc.), sus conceptos fundamentales (ficheros, protección de accesos, procesos, elementos de comunicación) y la funcionalidad de sus componentes (subsistemas para la gestión de procesos, memoria, entrada-salida).
GC1.47 Capacidad para seleccionar, diseñar e implementar nuevas estructuras de datos eficientes para la resolución de problemas.
GC1.48 Conocer las aplicaciones y beneficios de la gestión de proyectos en un entorno organizacional. Gestionar iniciativas a nivel de portafolio de proyectos. Saber cuándo aplicar un enfoque de gestión de proyectos ágil o predictivo. Entender la filosofía y fundamentos de la Agilidad.
GC1.49 Conocer los valores y principios del manifiesto ágil, en relación con las actividades empresariales. Utilizar los tableros Kanban como herramienta de trabajo práctica y efectiva. Armar tableros de gestión de tareas en distintos escenarios. Aplicar la metodología Scrum con sus equipos de trabajo. Gestionar proyectos ágiles de forma efectiva.



10. Programa de la asignatura

10.1- Objetivos docentes
Formar al estudiante en los principios fundamentales de metodologías ágiles y diseño de sistemas software, aplicar técnicas gestión de proyectos software. Potenciar la calidad en el desarrollo de sistemas software, facilitando que el software pueda ser usado por un amplio conjunto de usuarios. Introducir al alumno en el aprendizaje de la gestión de proyectos software: dirección, organización, planificación y gestión de proyectos informáticos. Adquirir capacidad de selección y aplicación de un catálogo de patrones de diseño en la construcción de una aplicación software, biblioteca o framework. Utilizar patrones de diseño para describir soluciones simples y de calidad a problemas específicos del diseño de software orientado a objetos. Reutilización de diseño software basada en patrones.
10.2- Unidades docentes (Bloques de contenidos)
Unidad 1 Valores y principios ágiles para el desarrollo del software - Desarrollo del software, características y modelos, aceptación del cambio. -Manifiesto ágil, repercusión del manifiesto en las metodologías de desarrollo, orígenes y fundamentos de las metodologías ágiles. -Gestión Ágil: Adaptación vs. Anticipación, ciclo de desarrollo ágil y resumen de metodologías (Scrum, Kanban, XP, FDD, etc.). Unidad 2 El marco de trabajo SCRUM - Introducción y Empirismo: El desarrollo incremental y el enfoque empírico. - Pilares y Valores de Scrum - El marco de trabajo: Roles (Product Owner, Scrum Master, Equipo de Desarrollo), Artefactos y Eventos (Planificación, Scrum Diario, Revisión, Retrospectiva). Unidad 3 Metodologías LEAN, KANBAN y SCRUMBAN - Técnicas Lean y Desarrollo de Software: Optimización del flujo, Value Stream Mapping, eliminación de desperdicios (Muda), y sistemas Pull. El desarrollo de software como proceso de fabricación y sus principios fundamentales. - Kanban: Motivaciones, visualizar el flujo de trabajo (tableros e ítems), establecer políticas explícitas, limitar el trabajo en progreso (WIP), y medir/optimizar el flujo (Lead time y Cycle time). - La metodología Scrumban: híbrido que mantiene los roles de Scrum y utiliza el flujo Kanban.



Unidad 4

Extreme Programming (XP)

- Valores, prácticas y ciclo de vida de XP
- Bienestar del Programador: Ritmo sostenible, cliente en el sitio y control de horas extra.

Prácticas

Prácticas

Desarrollo integral de un proyecto de aplicación web

- Control de Versiones y Entorno Colaborativo:
 - Git en proyectos web: Configuración del repositorio de la aplicación, flujos de trabajo colaborativos (estrategias de ramas, pull requests y revisiones de código).
- Desarrollo Base del Backend Web y Contenerización:
 - Frameworks web ágiles: Laravel (PHP) y creación de la arquitectura base (API/Backend) de la aplicación web del proyecto.
 - Despliegue portable: Empaquetado de la aplicación web y sus dependencias en un contenedor Docker para unificar los entornos de desarrollo y producción.
- Calidad del Código (TDD) y Automatización (CI):
 - Testing en aplicaciones web: Prácticas de Desarrollo Dirigido por Pruebas (TDD) aplicadas a los servicios y controladores web.
 - Integración Continua (CI): Principios y configuración de herramientas de integración continua para automatizar la construcción de la aplicación y la ejecución de las pruebas tras cada cambio en el código.
- Entrega Continua (CD) y Gestión Ágil del Proyecto:
 - Despliegue e Infraestructura: Scripts de despliegue, configuración de entornos de integración y despliegue de la aplicación web en la nube (Cloud).
 - Caso práctico integrador: Desarrollo iterativo de la aplicación web gestionando las tareas con un tablero Kanban (usando GitHub Projects), aplicando prácticas ágiles y manteniendo un flujo de integración y entrega continua completo (desde el commit hasta producción).

10.3- Acceso a la Bibliografía Recomendada desde Leganto

https://leganto.ubu.es/leganto/public/34BUC_UBU/lists?courseCode=8999&auth=SAML



11. Metodología de enseñanza y aprendizaje y su relación con las competencias/resultados de aprendizaje que debe adquirir/alcanzar el/la estudiante:

Metodología	Competencia/resulta do de aprendizaje relacionado	Horas presenciales	Horas de trabajo	Total de horas
Clases teóricas	CT1, CT5, CT9, GC1.25, GC1.49	24	0	24
Prácticas de laboratorio	CT1, CT3, CT7, CT12, CT17, GC1.47, GC1.49	24	0	24
Trabajo autónomo del alumno: Lecturas y recensiones. Resolución de problemas. Realización de trabajos, informes y memorias	CT1, CT3, GC1.25, GC1.48	0	96	96
Pruebas de evaluación	CT1, CT5	6	0	6
Total		54	96	150

12. Sistemas de evaluación:

1.Pruebas escritas: Mediante un cuestionario y con posibilidad de ejercicios cortos de modelado. 40%. Nota mínima de corte 4 sobre 10 puntos.
2.Bloque de Prácticas: 40%. Nota mínima de corte 4 sobre 10 puntos. Constará de Actividades Prácticas, las cuales se deben aprobar por separado. Cada una tendrá una nota de corte de 4 sobre 10.
3.Asistencia y participación en clase 20%

Procedimiento	Peso primera convocatoria	Peso segunda convocatoria
Realización de las prácticas	40 %	50 %
Asistencia a clase y participación	20 %	0 %
Pruebas de evaluación	40 %	50 %
Total	100 %	100 %



Evaluación excepcional:

No todos los procedimientos son recuperables en 2^{conv.} por la naturaleza del mismo. En el caso de que proceda la evaluación excepcional se realizan el mismo conjunto de pruebas en las fechas acordadas con los alumnos sujetos a dicha evaluación. En el caso de los alumnos que participen en el programa Universitario Cantera, la calificación se determinará en función del desempeño de las tareas que les sean asignadas en el marco del programa.

13. Recursos de aprendizaje y apoyo tutorial:

Se utilizará el campus virtual como soporte para las clases tanto teóricas como prácticas, así como para las consultas que los alumnos deseen realizar por este medio. Se utilizarán la pizarra y los Proyectoros. Se utilizarán las páginas Web relacionadas. Se utilizará la bibliografía disponible en la biblioteca. Se utilizarán las aplicaciones interactivas en la plataforma Ubuvirtual. Se proporcionará a los alumnos temas escritos y problemas, y tendrán la posibilidad de acceder a la bibliografía y al material específico. El horario de tutorías se indica al comienzo de la impartición de la asignatura.

14. Calendarios y horarios:

El calendario académico aprobado por la Junta de la Escuela Politécnica Superior. Los horarios y el calendario de exámenes finales se puede consultar en la página web de la titulación.

15. Idioma en que se imparte:

Español