



GUÍA DOCENTE

INGENIERÍA DEL SOFTWARE

DOBLE GRADO EN FÍSICA COMPUTACIONAL E INGENIERÍA DEL SOFTWARE

MODALIDAD: PRESENCIAL

CURSO ACADÉMICO: 2026-2027

Denominación de la asignatura:	Ingeniería del Software
Titulación:	Doble grado en Física Computacional e Ingeniería del Software
Facultad o Centro:	Centro Universitario de Tecnología y Arte Digital
Materia:	Ingeniería del Software
Curso:	4º
Cuatrimestre:	1
Carácter:	OB
Créditos ECTS:	6
Modalidad/es de enseñanza:	Presencial
Idioma:	Castellano
Profesor/a - email	Jorge Sanchez Cifuentes jorge.cifuentes@ext.live.u-tad.com
Página Web:	http://www.u-tad.com/

DESCRIPCIÓN DE LA ASIGNATURA

Descripción de la materia

Esta materia establece los conocimientos y técnicas necesarios para la correcta especificación, diseño e implementación de proyectos software atendiendo a las buenas prácticas y metodologías ingenieriles.

Descripción de la asignatura

La asignatura de Ingeniería del Software es la aquella en la que los alumnos aprenden a planificar y desarrollar proyectos software, pasando de concebir esta tarea desde el punto de vista del programador a contemplarla desde la perspectiva del director técnico de proyecto. En concreto, el alumno aprende a planificar y gestionar proyectos de desarrollo software mediante metodologías iterativas, a formar equipos de trabajo para sacar adelante proyectos de desarrollo software y a describir arquitecturas y diseños de software mediante lenguajes gráficos

COMPETENCIAS Y RESULTADOS DE APRENDIZAJE DE LA MATERIA

Competencias (genéricas, específicas y transversales)

COMPETENCIAS BÁSICAS Y GENERALES

CB1: Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

CB2: Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

CB3: Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.

CB4: Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.

CB5: Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

CG1 - Capacidad para entender, planificar y resolver problemas a través del desarrollo de soluciones informáticas.

CG2 - Desarrollo de soluciones informáticas respetuosas con el medio ambiente, los deberes sociales y los recursos naturales, además de cumplir con la legislación y la ética

CG3 - Conocimiento de los fundamentos científicos aplicables a la resolución de problemas informáticos

CG6 - Integración, como ingeniero del software, en entornos de trabajo multidisciplinarios demostrando capacidad de trabajo en equipo, versatilidad, flexibilidad, creatividad y respeto por el trabajo de los compañeros de otras áreas.

CG9 - Capacidad para aprender, modificar y producir nuevas tecnologías informáticas

CG10 - Uso de técnicas creativas para la realización de proyectos informáticos

COMPETENCIAS ESPECÍFICAS

CE10 - Capacidad para manejar un gestor de versiones de código y generar la documentación de una aplicación de forma automática.

CE18 - Capacidad para diseñar la arquitectura de una aplicación informática orientada a objetos empleando los patrones de diseño más adecuados e integrándolos en la arquitectura completa.

CE19 - Capacidad para concebir, diseñar a través de lenguajes gráficos e implementar una aplicación informática empleando distintas metodologías de desarrollo, desde la concepción del producto hasta su desarrollo final pasando por la definición de sus fases e iteraciones

CE20 - Capacidad para testar el funcionamiento y funcionalidad de una aplicación informática, elaborando planes de pruebas y empleando técnicas de diseño y programación orientado a las pruebas

CE22 - Conocimiento de las técnicas e implicaciones del mantenimiento de aplicaciones informáticas incluyendo aquellas que utilizan principios de ingeniería inversa para entender y modificar un software cuya estructura se desconoce

Resultados de aprendizaje

Al acabar la titulación, el graduado o graduada será capaz de:

- Conocer los lenguajes de especificación formal
- Ser capaz de identificar y usar patrones de diseño en la resolución de problema
- Manejar las técnicas de refactorización
- Entender el ciclo de vida del software
- Entender y aplicar las metodologías waterfall en el desarrollo
- Entender y aplicar Scrum en el desarrollo

CONTENIDO

Ciclos de vida del software

Metodologías iterativas de construcción de software

Metodologías ágiles de construcción de software

TEMARIO

BLOQUE 1. Introducción a la Ingeniería del Software

Tema 1. Introducción a la Ingeniería del Software

Tema 2. Riesgos y tema 3 Intro a los requisitos

BLOQUE 2. Gestión de requisitos y diseño Software

Tema 3. Introducción a la gestión de requisitos

Tema 4. Inception y Design Thinking

Tema 5. Diagramas UML

Tema 6. Visión & épicas e historias de usuario

Tema 7. Priorización y Estimación

BLOQUE 3. Modelos y metodologías de desarrollo Software

Tema 8. Modelos y metodologías de desarrollo del SW

Tema 9. Proceso unificado de desarrollo

Tema 10. SCRUM

Tema 11. KANBAN

Tema 12. XP

ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DOCENTES

Actividades formativas

Actividad Formativa	Horas totales	Horas presenciales
<i>Clases teóricas / Expositivas</i>	25	25
<i>Clases Prácticas</i>	29	29
<i>Tutorías</i>	4	2
<i>Estudio independiente y trabajo autónomo del alumno</i>	50	0
<i>Elaboración de trabajos (en grupo o individuales)</i>	32	0
<i>Actividades de Evaluación</i>	10	10

Metodologías docentes

Método expositivo o lección magistral

Aprendizaje de casos

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología Flipped classroom o aula invertida

Gamificación

Just in time Teaching (JITT) o aula a tiempo

Método expositivo o lección magistral

Método del caso

Aprendizaje basado en la resolución de problemas

Aprendizaje cooperativo o colaborativo

Aprendizaje por indagación

Metodología flipped classroom o aula invertida

Gamificación

DESARROLLO TEMPORAL

UNIDADES DIDÁCTICAS / TEMAS	PERÍODO TEMPORAL
-----------------------------	------------------

BLOQUE 1. Introducción a la Ingeniería del Software

Tema 1. Introducción a la Ingeniería del Software

Tema 2. Total Cost of Ownership

Tema 3 Intro a los requisitos

BLOQUE 2. Gestión de requisitos y diseño Software

Tema 3. Introducción a la gestión de requisitos

Tema 4. Inception y Design Thinking

Tema 5. Diagramas UML

Tema 6. Visión & épicas e historias de usuario

Tema 7. Priorización y Estimación

BLOQUE 3. Modelos y metodologías de desarrollo Software

Tema 8. Modelos y metodologías de desarrollo del SW

Tema 9. Proceso unificado de desarrollo

Tema 10. SCRUM

Tema 11. KANBAN

Tema 12. XP

SISTEMA DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	VALORACIÓN MÍNIMA RESPECTO A LA CALIFICACIÓN FINAL (%)	VALORACIÓN MÁXIMA RESPECTO A LA CALIFICACIÓN FINAL (%)
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	0	30
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	30	80
<i>Prueba Objetiva</i>	10	60

CRITERIOS ESPECÍFICOS DE EVALUACIÓN

ACTIVIDAD DE EVALUACIÓN	CONVOCATORIA ORDINARIA	CONVOCATORIA EXTRAORDINARIA
<i>Evaluación de la participación en clase, en prácticas o en proyectos de la asignatura</i>	10	10
<i>Evaluación de trabajos, proyectos, informes, memorias</i>	50	50
<i>Prueba Objetiva</i>	40	40

Consideraciones generales acerca de la evaluación

•1.- El proyecto final (con sus entregas intermedias) supondrá un 40% de la calificación definitiva. Es necesario superar este apartado con un 5 sobre 10 para aprobar la asignatura tanto en Convocatoria Ordinaria como Extraordinaria.

2.- No serán evaluados trabajos entregados fuera de plazo.

3.- El examen final supondrá un 50%. Es necesario superar el examen con un 5 sobre 10 para aprobar la asignatura tanto en Convocatoria Ordinaria como Extraordinaria.

4.- Sólo se podrá superar la asignatura con examen y proyecto final aprobados.

5.- En la Convocatoria Ordinaria, en el caso de tener una nota superior al 4 pero inferior al 5 en el examen, se valorará por el profesor si la asignatura se puede considerar aprobada en función de la calidad del proyecto final entregado y de la participación productiva del alumno en clase durante el curso.

6.- En la Convocatoria Ordinaria, en el caso de tener una nota superior al 4 pero inferior al 5 en el proyecto final, se valorará por el profesor si la asignatura se puede considerar aprobada en función de la nota del examen y de la participación productiva del alumno en clase durante el curso.

7.- “Participación activa” no significa venir a clase o “ganar” ejercicios colectivos. Se valora la resolución voluntaria de ejercicios y exposiciones. Asimismo, se valorará, como parte de este apartado, la exposición de ideas, la participación en debates, la presentación de propuestas o ejercicios adicionales y, en general, todo lo que demuestre una involucración en la asignatura, y no la mera asistencia pasiva.

8.- En el caso de que un alumno o alumna no haya conseguido alcanzar el 5 en proyecto y examen, pero tenga al menos un 4 en ambos, se le asignará un trabajo adicional especial a determinar que cubra los aspectos más deficientes de las entregas previas.

9.- Aquellos alumnos que suspendan el proyecto final o el examen en la Convocatoria Ordinaria tendrán la posibilidad de repetirlos en la Extraordinaria.

10.- Las notas de los exámenes y del proyecto no se guardan entre cursos académicos sucesivos.

11.- No es posible obtener Matrícula de Honor (MH) en la Convocatoria Extraordinaria.

12.- Cualquier escrito que el alumno presente (problemas, exámenes, comentarios de los programas, etc.) deberá estar bien presentado, correctamente redactado (con las comas, puntos y puntos y aparte en su lugar adecuado) y sin faltas ortográficas. La nota del escrito podrá bajar hasta un 20% en caso contrario, ya que a un universitario se le exige calidad máxima en su expresión escrita.

13.- En los exámenes no se permite el uso de apuntes ni de calculadoras de ningún tipo, para lo que el alumno debe remitirse a las instrucciones específicas del profesor sobre este tema.

14.- La asignatura COMPLETA estará suspensa si se descubre que un alumno ha copiado a otro (ambos estarán suspensos) o bien ha copiado de un libro o de Internet. Además, la universidad abrirá expedientes disciplinarios a ambos alumnos, pudiendo desembocar incluso en su expulsión.

15.- Se exige al menos un 80% de asistencia para aprobar la asignatura.

BIBLIOGRAFÍA / WEBGRAFÍA

Bibliografía básica

- Vliet, H. (2007); “SOFTWARE ENGINEERING: PRINCIPLES AND PRACTICES”; Editorial: Wiley.
- Pressman, R. S. (2010); “Ingeniería del software : Un enfoque practico” 7ª Ed.: McGraw-Hill
- Fowler, M. y Scott, K. (1999); “UML Distilled: A Brief Guide To The Standard Object Modeling Language”; Editorial: Addison Wesley;
- Beck, K. (1999); “eXtreme Programming explained”; Addison Wesley
- Lasa, C., Álvarez, A. y de las Heras, R. (2017); "Métodos Ágiles. Scrum, Kanban, Lean

Autor/a : Carmen Lasa Gómez, Alonso Álvarez García, Rafael de las Heras del Dedo"

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Tipología del aula

Aula teórica

Equipo de proyección y pizarra

Materiales:

Ordenador personal .

Cuaderno o tablet para tomar apuntes.

Software:

-