



GUÍA DOCENTE

INGENIERÍA DEL SOFTWARE

GRADO EN INGENIERÍA DEL SOFTWARE

MODALIDAD: A DISTANCIA

CURSO ACADÉMICO: 2023-2024

| | |
|--------------------------------|---|
| Denominación de la asignatura: | Ingeniería del Software |
| Titulación: | 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 de enseñanza: | A distancia |
| Idioma: | Castellano |
| Profesor / Email: | Alonso Alvarez / alonso.alvarez@ext.live.u-tad.com Carmen Lasa / carmen.lasa@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

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

TEMA 0. Presentación de la asignatura

TEMA 1. Introducción a la Ingeniería del Software

- 1.1. ¿Qué es la ingeniería del software?
- 1.2. El campo de acción de la Ingeniería del Software
- 1.3. Gestión de riesgos y la Ingeniería del Software
- 1.4. Orientación al valor y software como servicio

TEMA 2. Gestión de requisitos y diseño software

- 2.1. Necesidad de la gestión de requisitos
- 2.2. Análisis de requisitos
- 2.3. Requisitos no funcionales
- 2.4. Casos de uso
- 2.5. Diagramas de clases
- 2.6. Diagramas de secuencia y comunicación
- 2.7. Inception y Design Thinking
- 2.8. Épicas y US

2.9. Priorización y estimación

2.10. Gestión de stakeholders

TEMA 3. Modelos y metodologías de desarrollo del software

3.1. Introducción al proceso de desarrollo software

3.2. Modelos de desarrollo software

3.3. Metodologías

3.4. Metodologías pesadas UDP/PUD

3.5. Metodologías ligeras. Agile

3.6. Scrum

3.7. Kanban

3.8. XP y otras formas de aplicar agilidad

3.9. Escalado

ACTIVIDADES FORMATIVAS Y METODOLOGÍAS DE APRENDIZAJE

Actividades formativas

| Actividad Formativa | Horas totales | Horas síncronas |
|--|---------------|-----------------|
| <i>Sesiones teóricas virtuales síncronas</i> | 2,00 | 2 |
| <i>Sesiones teóricas virtuales asíncronas</i> | 21,00 | 0 |
| <i>Sesiones prácticas virtuales síncronas</i> | 4,00 | 4 |
| <i>Sesiones prácticas virtuales asíncronas</i> | 17,00 | 0 |
| <i>Debate y discusión oral y/o escrita.</i> | 8,00 | 0 |
| <i>Tutorías</i> | 4,00 | 4 |
| <i>Estudio independiente y trabajo autónomo del alumno</i> | 48,00 | 0 |
| <i>Elaboración de trabajos (en grupo o individuales)</i> | 30,00 | 0 |
| <i>Actividades de Evaluación</i> | 10,00 | 10 |
| <i>Test de autoevaluación</i> | 6,00 | 0 |
| <i>Prácticas externas</i> | 0,00 | 0 |

| | | |
|--|------------|-----------|
| <i>Preparación y defensa virtual del TFG</i> | 0,00 | 0 |
| <i>Seguimiento de proyectos</i> | 0,00 | 0 |
| TOTAL | 150 | 20 |

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

Presentación - semana 1

Unidad 1 - semana 2-3

Unidad 2 - semana 4-5

Unidad 3 - semana 6-7

Unidad 4 - semana 7-8

Unidad 5 - semana 9-10

Unidad 6 - semana 11-12

Repaso - semana 13-14

Evaluación - semana 15

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> | 10 | 20 |
| <i>Evaluación de trabajos, proyectos, informes, memorias</i> | 10 | 20 |
| <i>Prueba Objetiva</i> | 60 | 70 |

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> | 20 | 10 |
| <i>Evaluación de trabajos, proyectos, informes, memorias</i> | 20 | 20 |
| <i>Prueba Objetiva</i> | 60 | 70 |

Consideraciones específicas acerca de la evaluación

Será necesario que obtener una nota mínima de 4 puntos (sobre 10) en la prueba final presencial para que se realice la media con las actividades formativas.

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

Bibliografía recomendada

- 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

MATERIALES, SOFTWARE Y HERRAMIENTAS NECESARIAS

Materiales:

Ordenador personal .

Cuaderno o tablet para tomar apuntes.

Software:

-