

Curso Académico: (2021 / 2022)

Fecha de revisión: 30-06-2021

Departamento asignado a la asignatura: Departamento de Informática

Coordinador/a: LOPEZ CUADRADO, JOSE LUIS

Tipo: Obligatoria Créditos ECTS : 6.0

Curso : 2 Cuatrimestre : 2

REQUISITOS (ASIGNATURAS O MATERIAS CUYO CONOCIMIENTO SE PRESUPONE)

Programación (1º,1)

Estructura de datos y Algoritmos (1º,2)

Ingeniería del Software (2º,1)

Teoría de Autómatas y Lenguajes Formales (2º,1)

OBJETIVOS

El objetivo del curso es permitir al estudiante profundizar en prácticas ágiles de desarrollo que facilitan la obtención de componentes software con alta calidad.

Los resultados del aprendizaje serán los siguientes:

R1. Conocimiento y comprensión: Tener conocimientos básicos y la comprensión de los fundamentos científicos y tecnológicos de la Ingeniería Informática, así como un conocimiento específicos de las ciencias de la computación, la ingeniería de computadores y sistemas de información.

R2 Análisis de la Ingeniería: Ser capaces de identificar problemas de Ingeniería Informática, reconocer sus especificaciones, establecer diferentes métodos de resolución y seleccionar el más adecuado para su solución, teniendo en cuenta las limitaciones sociales, salud humana, Medio Ambiente, y comerciales aplicables en cada caso.

R3 Diseño en Ingeniería: Ser capaces de realizar diseños de ingeniería de acuerdo a su nivel de conocimiento y comprensión que cumplan con las especificaciones requeridas colaborando con otros ingenieros y titulados. El diseño abarca dispositivos, procesos, métodos y objetos, y especificaciones más amplias que las estrictamente técnicas, lo cual incluye conciencia social, salud y seguridad, y consideraciones medioambientales y comerciales.

Competencias Básicas y Generales

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

CG5 - Usar herramientas informáticas, de propósito general, colaborativas y de optimización del trabajo para la planificación e

implementación efectiva de proyectos.

CG9 - Utilizar de forma eficiente medios TIC para redactar informes técnicos y memorias de proyectos y trabajos sobre

Informática, así como presentaciones de calidad.

CGO3 - Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas,

servicios y aplicaciones informáticas, así como de la información que gestionan

CGO5 - Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de

la ingeniería del software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos

CGO7 - Capacidad para conocer, comprender y aplicar la legislación necesaria durante el desarrollo de la profesión de Ingeniero

Técnico en Informática y manejar especificaciones, reglamentos y normas de obligado cumplimiento

Competencias específicas

CECRI1 - Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas

informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente
CECRI8 - Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
CECRI16 - Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

DESCRIPCIÓN DE CONTENIDOS: PROGRAMA

- 1.- Aspecto éticos y legales en la profesión de Ingeniero de Software.
 - 1.1.- La profesión de Ingeniero de Software
 - 1.2.- Código ético de la profesión de Ingeniero de Software.
- 2.- Prácticas genéricas del desarrollo ágil
 - 2.1.- El proceso de desarrollo de software.
 - 2.2.- Programación en Parejas
 - 2.3.- Estándares de codificación
 - 2.4.- Propiedad Colectiva de Código
- 3.- Desarrollo Dirigido por Pruebas
 - 3.1.- Principios del Desarrollo Dirigido por Pruebas
 - 3.2.- Técnicas de Prueba Funcionales
 - 3.3.- Técnicas de Prueba Estructurales
 - 3.4.- Automatización de Pruebas Unitarias
 - 3.5.- Integración Continua Automatizada
- 4.- Refactoring y Diseño Simple
 - 4.1.- Refactoring
 - 4.2.- Principios de Diseño Simple
 - 4.3.- Patrones de Diseño para la Asignación de Responsabilidades

ACTIVIDADES FORMATIVAS, METODOLOGÍA A UTILIZAR Y RÉGIMEN DE TUTORÍAS

- Clases Teóricas: 1,5 ECTS. Tienen por objetivo alcanzar las competencias específicas cognitivas y procedimentales de la asignatura
- Clases Prácticas: 1,5 ECTS. Desarrollan las competencias específicas instrumentales y la mayor parte de las transversales, s. También tienen por objetivo desarrollar las capacidades específicas actitudinales. Consisten en el diseño y desarrollo de un componente de software. En el ámbito de esta actividad de desarrollo se ejercitarán las técnicas de especificación de requisitos, diseño, revisión de calidad y pruebas del código que implementa el componente. Estas actividades se realizarán en equipo. Todo ello desarrollado conforme a los aspectos éticos y legales de desarrollo del proceso.
- Trabajos Prácticos 2 ECTS: elaborados con y sin presencia del profesor sobre un caso de proyecto de desarrollo de un componente de software, donde se profundice sobre todos los aspectos teóricos abordados en la materia aspecto de la materia.
- Tutorías: Asistencia individualizada (tutorías individuales) o en grupo (tutorías colectivas) a los estudiantes por parte del profesor.
- Examen (Practica Final): 1 ECTS. Tiene por objeto incidir y complementar en el desarrollo de las capacidades específicas cognitivas y procedimentales.

SISTEMA DE EVALUACIÓN

Los ejercicios y exámenes además de servir como actividad formativa tienen el doble objetivo de ser medida para el sistema de evaluación. El sistema de evaluación incluye la valoración de las actividades académicas dirigidas y prácticas según la siguiente ponderación.

Ejercicios Guiados (Evaluación Continua): 30%
Tests/Ejercicios de Teoría (Evaluación Continua): 30%
Práctica Final (Examen): 40%

Para superar la asignatura, es imprescindible aprobar, por separado, cada una de las partes (Teoría, Ejercicios Guiados y Práctica Final).

Es imprescindible superar, por separado, cada una de las partes de la asignatura para poder superarla a nivel global.

Peso porcentual del Examen Final:	40
Peso porcentual del resto de la evaluación:	60

BIBLIOGRAFÍA BÁSICA

- Beck, Ken, et al.. Test-Driven Development By Example. , Three Rivers Institute., 2002
- Beck, Ken. Una explicación de la Programación Extrema, Addison-Wesley, 2000
- Craig S. Larman Applying UML and Patterns., Pearson Education . 3er Edition, 2012
- Fowler, Martin et al.. Refactoring: Improving the Design of Existing Code. , Addison-Wesley. , 1999
- Lee Copeland. A Practitioner's Guide to Software Test Design., Artech House Publishers, 2003

BIBLIOGRAFÍA COMPLEMENTARIA

- Paul C. Jorgensen Software Testing: a craftsman's approach. , CRC.
- Roger S. Pressman. Ingeniería del Software. Un enfoque práctico. , McGraw Hill. 7ª Edición., 2009
- Spyros Xanthakis, Michel Maurice, Antonio de Amescua, Olivier Houri, Luc Griffet. Test and contrôle des logiciels : méthodes, techniques and outils, EC2..