Universidad Zaragoza

Curso Académico: 2022/23

30391 - Análisis y diseño de software

Información del Plan Docente

Año académico: 2022/23
Asignatura: 30391 - Análisis y diseño de software
Centro académico: 110 - Escuela de Ingeniería y Arquitectura
Titulación: 581 - Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación
Créditos: 6.0
Curso: 4
Periodo de impartición: Primer semestre
Clase de asignatura: Optativa
Materia:

1. Información Básica

1.1. Objetivos de la asignatura

La asignatura y sus resultados previstos responden a los siguientes planteamientos y objetivos:

Tras haber cursado con aprovechamiento un conjunto de asignaturas donde se han aprendido conceptos de programación (Fundamentos de Informática, Programación de Redes y Servicios) el alumno se introduce ahora en el paradigma orientado a objeto. En esta asignatura va a aprender a abordar la construcción de software de mayor entidad y desde una perspectiva global, no solamente centrada en aspectos de implementación. Tomando como punto de partida la descripción de un software a construir, el alumno aprenderá a determinar sus requisitos, analizarlos, realizar un diseño, y plantear un conjunto de pruebas sobre el software construido.

La asignatura tiene un marcado carácter aplicado. El alumno aprenderá los conceptos necesarios sobre análisis, diseño y pruebas aplicándolos a un conjunto de ejemplos de sistemas presentados tanto en las clases de problemas, como en las distintas sesiones de las prácticas de laboratorio.

Estos planteamientos y objetivos están alineados con el Objetivo 9: Industria, innovación e infraestructuras, de la Agenda 2030 de Naciones Unidas (https://www.un.org/sustainabledevelopment/es/), de tal manera que la adquisición de los resultados de aprendizaje de la asignatura proporciona capacitación y competencia para contribuir en cierta medida a su logro.

1.2. Contexto y sentido de la asignatura en la titulación

La asignatura de Análisis y Diseño Software forma parte de la materia denominada Diseño de Servicios Telemáticos que cubre competencias obligatorias dentro de la titulación del grado en Ingeniería de Tecnologías y Servicios de Telecomunicación en la tecnología específica de Telemática.

En concreto, esta asignatura complementa la formación en desarrollo de software dentro de la tecnología específica de Telemática. Un ingeniero graduado en esta especialidad se enfrenta casi a diario con el reto de desarrollar soluciones software en dominios heterogéneos: redes, procesado de señales, sistemas digitales, etc. Esta asignatura le proporcionará las bases para afrontar esos desarrollos con una metodología orientada a objetos.

1.3. Recomendaciones para cursar la asignatura

El alumno que curse esta asignatura contará con una formación en programación del nivel que se adquiere en la asignatura de Programación de Redes y Servicios.

2. Competencias y resultados de aprendizaje

2.1. Competencias

Al superar la asignatura, el estudiante será más competente para...

Concebir, diseñar y desarrollar proyectos de Ingeniería (C1).

Planificar, presupuestar, organizar, dirigir y controlar tareas, personas y recursos (C2).

Usar las técnicas, habilidades y herramientas de la Ingeniería necesarias para la práctica de la misma (C6).

Combinar los conocimientos generalistas y los especializados de Ingeniería para generar propuestas innovadoras y competitivas en la actividad profesional (C3).

Resolver problemas y tomar decisiones con iniciativa, creatividad y razonamiento crítico (C4).

Aplicar las tecnologías de la información y las comunicaciones en la Ingeniería (C11).

Diseñar arquitecturas de redes y servicios telemáticos (CT6)

Programación de servicios y aplicaciones telemáticas, en red y distribuidas (CT7).

2.2. Resultados de aprendizaje

El estudiante, para superar esta asignatura, deberá demostrar los siguientes resultados...

Entiende el ciclo de vida del software, conoce la existencia de diferentes modelos de ciclo de vida y sabe en qué circunstancias aplicar cada uno de ellos.

Sabe distinguir los conceptos de la programación estructurada de los de la programación orientada a objetos, y de un modo más general entiende el paradigma orientado a objetos desde el análisis hasta la implementación.

Es capaz de realizar la captura de requisitos del software: funcionales y no funcionales.

Es capaz de realizar el análisis de un sistema software distinguiendo entre el modelo estático y el dinámico.

Es capaz de realizar diseños software orientados a objetos.

Conoce y sabe aplicar diferentes patrones de diseño y distinguir distintos patrones arquitecturales.

Conoce y sabe utilizar el lenguaje de modelado unificado (UML) tanto en el análisis como en el diseño del software.

Conoce los fundamentos básicos de las pruebas del software.

Programa software con librerías de acceso a bases de datos relacionales: ODBC.

Conoce el paradigma de Invocación Remota de Objetos y los conceptos arquitecturales subyacentes

Concibe el proyecto software en su totalidad, desde la captura de requisitos a la implementación, y es capaz de crear la documentación necesaria para cada una de las etapas del ciclo de vida.

2.3. Importancia de los resultados de aprendizaje

Aprender a abordar el desarrollo de sistemas software aplicando los principios de una ingeniería es esencial para un graduado en ingeniería de servicios y tecnologías de telecomunicación dentro de la tecnología específica de Telemática que participe en un equipo humano de desarrollo de software de tamaño medio o grande. Lo que aprenda en esta asignatura, que complementa a lo ya aprendido en las asignaturas de programación, será un importante paso adelante en su formación como ingeniero.

3. Evaluación

3.1. Tipo de pruebas y su valor sobre la nota final y criterios de evaluación para cada prueba

El estudiante deberá demostrar que ha alcanzado los resultados de aprendizaje previstos mediante las siguientes actividades de evaluación

Convocatoria de Junio. La evaluación de la asignatura se realiza en base a tres pruebas:

P1. Ejercicios de cada uno de los temas desarrollados en las clases de teoría y problemas. De forma individual o en grupo trabajará y posteriormente entregará, en la fecha acordada con el profesor, los ejercicios propuestos para el tema. Se requiere una nota mínima de 5.0 puntos en esta prueba para aprobar la asignatura. Si se obtiene esta nota mínima, entonces la prueba pondera un 45% en la nota de la asignatura y, si no se alcanza este mínimo, entonces la calificación en la asignatura es la de esta prueba.

P2. Resultados de prácticas de laboratorio. Esta prueba pondera un 35% en la nota de la asignatura. Cada una de las 6 sesiones de prácticas será evaluada. Cada grupo será evaluado bien durante la sesión de laboratorio bien mediante la entrega del trabajo que se indiquen en la sesión de prácticas. Es obligatorio superar esta evaluación obteniendo una nota mínima de 5.0 puntos para aprobar la asignatura.

P3. Prueba escrita. Consistirá en la resolución de problemas y preguntas teóricas sobre la materia desarrollada a lo largo del curso. Esta prueba pondera un 20% en la nota de la asignatura. Es obligatorio superar esta evaluación obteniendo una nota mínima de 5.0 puntos para aprobar la asignatura.

Convocatoria segunda. La evaluación de la asignatura se realiza en base a tres pruebas análogas a las de la convocatoria de Junio, con las mismas ponderaciones y exigencia de notas mínimas. Las calificaciones del alumno obtenidas en la convocatoria de Junio en cualquiera de las pruebas (P1, P2, P3) se mantienen, salvo que el alumno opte por presentarse a la prueba correspondiente en esta nueva convocatoria, en cuyo caso prevalecerá la nueva calificación.

4. Metodología, actividades de aprendizaje, programa y recursos

4.1. Presentación metodológica general

El proceso de aprendizaje que se ha diseñado para esta asignatura se basa en lo siguiente:

1. El estudio y trabajo continuado desde el primer día de clase.
2. El aprendizaje de conceptos y metodologías para el análisis, diseño, verificación y validación de software a través de las clases magistrales, en las que se favorecerá la participación de los alumnos.
3. La aplicación de tales conocimientos al análisis, diseño, verificación y validación de ejemplos de sistemas software en las clases de problemas. En estas clases los alumnos desempeñarán un papel activo en la discusión y resolución de los problemas. En algunas de estas clases el trabajo de cada alumno será evaluado, mediante la realización de pruebas evaluables con carácter voluntario.
4. Las clases de prácticas de laboratorio en las que el alumno aprenderá la tecnología necesaria para realizar el análisis, diseño, verificación y validación de una aplicación informática a lo largo de distintas sesiones.

Trabajo del estudiante

La dedicación del estudiante para alcanzar los resultados de aprendizaje en esta asignatura se estima en 150 horas distribuidas del siguiente modo:

- 60 horas, aproximadamente, de actividades presenciales (sesiones teóricas -25- y de problemas -15- y sesiones de laboratorio -20-)
- 24 horas de trabajo en grupo
- 60 horas de trabajo y estudio individual efectivo (estudio de apuntes y textos, resolución de problemas, preparación clases y prácticas, desarrollo de programas)
- 6 horas dedicadas a distintas pruebas de evaluación

4.2. Actividades de aprendizaje

El programa que se ofrece al estudiante para ayudarle a lograr los resultados previstos comprende las siguientes actividades...

En las clases magistrales se desarrollará el temario de la asignatura.

En las clases de problemas se resolverán problemas de aplicación de los conceptos y técnicas presentadas en el programa de la asignatura.

En las sesiones de prácticas cada alumno deberá realizar, individualmente o en equipo, trabajos directamente relacionados con los temas estudiados en la asignatura.

4.3. Programa

- Introducción a la ingeniería del software: ciclo de vida del software
- Requisitos del software
- Análisis orientado a objetos: modelado estático, modelado dinámico
- Diseño orientado a objetos: Patrones de diseño
- Pruebas del producto software
- Objetos distribuidos

4.4. Planificación de las actividades de aprendizaje y calendario de fechas clave

Calendario de sesiones presenciales y presentación de trabajos

La organización docente de la asignatura prevista es la siguiente:

- Clases teóricas y de problemas (2 horas semanales durante 5 semanas y 3 horas semanales durante 10 semanas).
- Clases prácticas de laboratorio (seis sesiones de 3 horas cada dos semanas). Son sesiones de trabajo análisis, diseño, programación y pruebas de software en laboratorio, tuteladas por un profesor, en las que participan los alumnos de cada uno de los subgrupos en los que se divide el grupo.

El calendario de exámenes y las fechas de entrega de trabajos se anunciará con suficiente antelación

4.5. Bibliografía y recursos recomendados

http://psfunizar10.unizar.es/br13/egAsignaturas.php?codigo=30391