

Grado en Ingeniería Informática

30245 - Arquitectura software

Guía docente para el curso 2014 - 2015

Curso: 3, Semestre: 2, Créditos: 6.0

Información básica

Profesores

- **Rubén Bejar Hernández** rbejar@unizar.es
- **José Javier Meseguer Hernaiz** jmerse@unizar.es

Recomendaciones para cursar esta asignatura

El alumno que curse esta asignatura ha de contar con una formación como la que se recibe en la asignatura de "Ingeniería del Software".

Actividades y fechas clave de la asignatura

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

Inicio

Resultados de aprendizaje que definen la asignatura

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

- 1:** Sabe diseñar y documentar la arquitectura de un sistema de software, desde distintos puntos de vista y niveles de abstracción.
- 2:** Conoce diferentes patrones arquitecturales y es capaz de aplicarlos para diseñar sistemas software.
- 3:** Es capaz de proponer patrones arquitecturales originales aplicables al diseño de sistemas software.
- 4:** Sabe aplicar técnicas aprendidas en "Ingeniería de Software" (casos de uso, modelado estructural y modelado dinámico) al diseño de arquitecturas software
- 5:**

Conoce y utiliza herramientas CASE para diseñar y documentar arquitecturas software.

6:

Conoce la existencia de diferentes metodologías de arquitectura de software y las diferencias más importantes entre ellas.

Introducción

Breve presentación de la asignatura

Todo sistema software tiene una estructura fundamental, su arquitectura. En sistemas software de tamaño mediano, grande o muy grande, diseñar y documentar su arquitectura es condición necesaria para llevar a cabo con éxito su construcción, así como para realizar diferentes tipos de análisis (prestaciones, fiabilidad, escalabilidad, evolución, etc.). La arquitectura de un sistema software está formada por sus elementos fundamentales, las propiedades visibles de los mismos y las relaciones que entre ellos existen. Esta asignatura busca que los alumnos conozcan y apliquen técnicas de diseño y documentación específicas para arquitecturas de software. Para ello se presentará una propuesta para documentar las arquitecturas software y varios patrones arquitecturales. Los alumnos aplicarán el conocimiento adquirido al diseño de la arquitectura de varios sistemas software.

Contexto y competencias

Sentido, contexto, relevancia y objetivos generales de la asignatura

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

Tras haber cursado la asignatura de “Ingeniería del Software”, los alumnos saben aplicar técnicas de análisis y diseño orientados a objetos. En esta asignatura los alumnos van a aprender a ver los sistemas de software desde un punto de vista más general, más centrado en sus estructuras fundamentales que en los detalles de sus componentes. Las técnicas y metodologías que van a aprender y aplicar se basan en las que aprenden en “Ingeniería del Software”, pero están concebidas para abordar el diseño y documentación de estos aspectos más generales de los sistemas de software.

La asignatura tiene un marcado carácter práctico, no limitándose a ser un catálogo teórico de técnicas y patrones arquitecturales. Los alumnos aprenderán los conceptos necesarios sobre arquitecturas de software aplicándolos a un conjunto de casos de estudio, tanto en las clases de problemas como en el trabajo de asignatura. En este trabajo, desarrollado en equipo, los conceptos se aplicarán a un proyecto de software que podrá ser desarrollado conjuntamente con otras asignaturas de la intensificación (como Ingeniería de Requisitos o Verificación y Validación) o de la materia común de Ingeniería del Software (asignatura de Proyecto Software).

Contexto y sentido de la asignatura en la titulación

La asignatura “Arquitecturas Software” proporciona a los alumnos una introducción a algunas de las propuestas, técnicas y patrones necesarios para la construcción de sistemas informáticos de tamaño mediano, grande y muy grande, a partir de los conocimientos adquiridos fundamentalmente en la asignatura de “Ingeniería del Software”. Es una asignatura obligatoria dentro de la especialidad en “Ingeniería del Software”.

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

1:

Concebir, diseñar y desarrollar proyectos de Ingeniería.

2:

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

3:

Comunicar y transmitir conocimientos, habilidades y destrezas en castellano y en inglés.

4: Usar las técnicas, habilidades y herramientas de la Ingeniería necesarias para la práctica de la misma.

5: Desarrollar, mantener y evaluar servicios y sistemas software que satisfagan todos los requisitos del usuario y se comporten de forma fiable y eficiente, sean asequibles de desarrollar y mantener y cumplan normas de calidad, aplicando las teorías, principios, métodos y prácticas de la Ingeniería del Software.

6: Identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales.

Importancia de los resultados de aprendizaje que se obtienen en la asignatura:

Conocer las técnicas específicas de arquitectura de software facilita concebir y llevar a cabo sistemas de software, especialmente cuando son medianos, grandes o muy grandes. Este tipo de técnicas ayudan a diseñar y documentar sistemas de software mantenibles, eficientes y escalables, aplicando los patrones arquitecturales más adecuados para cada caso. Los conocimientos de arquitectura de software permitirán a los alumnos crecer como ingenieros, siendo claves para enlazar sus conocimientos previos sobre programación e ingeniería del software con el desarrollo profesional de sistemas de software grandes.

Evaluación

Actividades de evaluación

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

1: **Convocatoria de Junio.** La evaluación de la asignatura se realiza en base a dos pruebas:

P1. Prueba escrita en la que responder cuestiones y resolver ejercicios y problemas. 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 40% en la nota de la asignatura.

P2. Trabajo de asignatura y resultados de prácticas. Es obligatorio presentar este trabajo y obtener una nota mínima de 5.0 puntos para aprobar la asignatura. Si se obtiene esta nota mínima, entonces esta prueba pondera un 60% en la nota de la asignatura. Cada grupo deberá entregar y defender un trabajo focalizado en las diferentes vistas de la arquitectura de un software realizado durante el curso y los resultados obtenidos en las sesiones de prácticas.

Es obligatorio realizar y entregar ambas pruebas para poder aprobar la asignatura. Si en una prueba, o en las dos, la nota obtenida fuera inferior a 5.0, la calificación final de la asignatura será la media ponderada de las dos calificaciones (40% P1 y 60% P2), con un máximo de 4.0.

2: **Convocatoria de Septiembre.** La evaluación de la asignatura se realiza en base a dos 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 cualquier de las pruebas (P1 y P2) se mantienen en Septiembre, salvo que el alumno opte por presentarse a la prueba correspondiente en esta nueva convocatoria, en cuyo caso prevalecerá la nueva calificación.

Actividades y recursos

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 diseño y documentación de arquitecturas de software a través de las clases magistrales, en las que se favorecerá la participación de los alumnos. La aplicación de tales conocimientos al diseño y documentación de la arquitectura de aplicaciones de ejemplo en las clases de problemas. En estas clases los alumnos desempeñarán un papel activo en la discusión y resolución de los problemas.
3. El trabajo en equipo, donde los alumnos desarrollarán, bajo la supervisión de los profesores, un trabajo focalizado en las diferentes vistas de la arquitectura de un software realizado durante el curso.

Actividades de aprendizaje programadas (Se incluye programa)

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

- 1:**
En las clases impartidas en el aula se desarrollará el temario de la asignatura.
- 2:**
En las clases de problemas se resolverán problemas de aplicación de los conceptos y técnicas presentadas en el programa de la asignatura.
- 3:**
Se realizará un trabajo que consistirá en diseñar y documentar la arquitectura de diferentes sistemas software.

Planificación y calendario

Calendario de sesiones presenciales y presentación de trabajos

La organización docente prevista de la asignatura es la siguiente.

- Clases teóricas (2 horas semanales)
- Clases de problemas (1 hora semanal)
- Tutorías de los trabajos en grupo. Sesiones de 2 horas cada semana: son sesiones de diseño y documentación de las distintas vistas de la arquitectura de un software, tuteladas por los profesores. Además se establecerá seguimiento de los trabajos cada vez que se considere oportuno.

Programa

Programa de la asignatura

- Introducción a la arquitectura de software.
- Documentación de la arquitectura software
 - Vista de Módulos.
 - Vista de Componente y Conector.
 - Vista de Distribución.
 - Documentación general y de interfaces.
 - Vista Enterprise.
- Patrones arquitecturales
 - MVC
 - Cliente-servidor
 - SOA
 - Broker
 - Publish-subscribe
 - Pizarra

- Shared-data
- Arquitecturas adaptativas

Trabajo

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:

- 45 horas, aproximadamente, de actividades presenciales (sesiones en el aula teóricas -30- y de problemas -15-)
- 65 horas de trabajo en grupo
- 35 horas de trabajo y estudio individual efectivo (estudio de apuntes y textos, resolución de problemas, y preparación de clases y prácticas, desarrollo de programas)
- 5 horas dedicadas a distintas pruebas de evaluación

Bibliografía

Bibliografía recomendada

- Paul Clements, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Paulo Merson, Robert Nord and Judith Stafford. 2010. Documenting Software Architectures: Views and Beyond, Second Edition. Addison-Wesley Professional.
- Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal. 1996. Pattern-Oriented Software Architecture. A System of Patterns. John Wiley & Sons.

Material adicional

- Peter F. Linington, Zoran Milosevic, Akira Tanaka, Antonio Vallecillo. 2012. Building Enterprise Systems with ODP. CRC Press..

Referencias bibliográficas de la bibliografía recomendada

Escuela Universitaria Politécnica

- Building Enterprise Systems with ODP / Peter F. Linington (et al.). CRC Press, 2012
- Documenting Software Architectures: Views and Beyond / Paul Clements... (et al.). Addison-Wesley Professional, 2010
- Pattern-Oriented Software Architecture. A system of Patterns / Frank Buschmann (et al.). Ireland : John Wiley & Sons, 1996

Escuela Politécnica Superior

- Building Enterprise Systems with ODP / Peter F. Linington, [et al.]. CRC Press, 2012
- Documenting Software Architectures: Views and Beyond / Paul Clements... [et al.]. - 2nd ed. Addison-Wesley Professional, 2010
- Pattern-Oriented Software Architecture. A System of Patterns / Frank Buschmann... [et al.]. John Wiley & Sons, 1996