

ARQUITECTURAS PARA SISTEMAS SOFTWARE

Curso 2011/2012

(Código:31105039)

1.PRESENTACIÓN

Un informe publicado en 2004 por el Standish Group^[1] revelaba que sólo el 29% de los proyectos software satisfacen los requisitos de los clientes, se entregan a tiempo y se ajustan a los presupuestos. El 53% acaban incumpliendo algún requisito, se entregan fuera de plazo o cuestan más de lo presupuestado. El 18% se cancelan antes de su finalización o jamás llegan a utilizarse. Sin duda, aún queda mucho por hacer en el desarrollo de software.

En los últimos 40 años, se han producido grandes avances en el terreno de la codificación: creación de compiladores e intérpretes eficientes, desarrollo de la algoritmia, aparición de nuevos paradigmas de programación, utilización de los sistemas de tipos para la prevención de errores... Sin embargo, a medida que crece el tamaño de las aplicaciones informáticas, se hace más evidente la necesidad de organizarlas correctamente. Es decir, la importancia de la codificación disminuye frente a la relevancia del diseño de software.

El presente curso pretende transmitir al alumno la enorme importancia del diseño en el desarrollo de software y ofrecerle una panorámica sobre las técnicas de diseño más aplicadas actualmente y, con mayor probabilidad, en el futuro. A parte del planteamiento introductorio sobre la relevancia del diseño y cómo afecta al ciclo de vida y a la propia ejecución del desarrollo de Software, se presentan los conceptos de Estilos Arquitectónicos y Directrices Arquitectónicas, haciendo un recorrido por los estilos más relevantes. El núcleo principal de la asignatura está constituido por el diseño arquitectónico y el diseño detallado. En lo que se refiere al diseño arquitectónico, su estudio se ilustra en el ámbito de las Arquitecturas Dirigidas por Modelos (MDA) por ser una propuesta en desarrollo, prometedora y con una buena proximidad al formalismo. En cuanto al diseño detallado, se utilizan los '*patrones de diseño*' por su potencial para que el estudiante adquiriera la madurez para resolver nuevos problemas de diseño a partir de soluciones eficaces y genéricas -los patrones-.

^[1] The Standish Group. 2004 CHAOS Demographics and Project Resolution. <http://www.standishgroup.com>

2.CONTEXTUALIZACIÓN

La asignatura "Arquitecturas para Sistemas Software" se encuentra integrada en el Máster en Ingeniería de Sistemas y Control dentro del Módulo denominado "Ingeniería de Software". Este módulo está organizado en dos materias y se desarrolla en un total de seis asignaturas anuales y optativas. Concretamente la asignatura "Arquitecturas para Sistemas Software" es una de las cinco asignaturas que forman la materia "Ingeniería del Desarrollo de Software". Las otras cuatro asignaturas son "Generación Automática de Código", "Especificación de los Sistemas Software", "Desarrollo de Líneas de Producto Software mediante un Enfoque Generativo" y "Arquitecturas Orientadas a Servicios". La otra materia del módulo es "Ingeniería de la Gestión del Software" que se compone de una sola asignatura, también anual, denominada "Gestión y Mejora de Procesos Software". Por otro lado, toda la materia "Ingeniería del Desarrollo de Software" está íntimamente relacionada –y se complementa– con la materia "Ingeniería de la Gestión del Software" y con la asignatura "Gestión y Mejora de Procesos Software"; puesto que se dirige a la gestión y mejora de los procesos involucrados en el desarrollo de Software, objeto de esta asignatura y materia.

También es interesante resaltar que, en cualquiera de las asignaturas del Módulo "Ingeniería de Sistemas Informáticos", se ofrecen campos de desarrollo de Software diferenciados y específicos para aplicar y ejercitar las conclusiones, enseñanzas y capacidades adquiridas en este Módulo.

3.REQUISITOS PREVIOS RECOMENDABLES



La formación previa que deberían tener los alumnos para el adecuado seguimiento de esta asignatura son los propios de ingreso al posgrado, haciendo especial recomendación en conocimientos matemáticos y tecnológicos en materias de estadística, algoritmia, programación, especificación formal y orientación a objetos, así como en la lectura y comprensión de textos en inglés técnico.

4.RESULTADOS DE APRENDIZAJE

Los resultados de aprendizaje que se espera alcanzar con esta asignatura por parte del estudiante son:

- Conocer los Estilos arquitectónicos, Arquitecturas y Directrices arquitectónicas de desarrollo de Software
- Saber aplicar un estilo arquitectónico a un problema de desarrollo, diseñar con componentes o construir patrones y utilizarlos en el diseño
- Analizar herramientas, entornos y aplicaciones de desarrollo y evaluar cuál es la más adecuada para el problema planteado
- Capacidad para desarrollar Software mejorando su calidad y el rendimiento productivo
- Saber aprovechar Técnicas, Metodologías y Arquitecturas de Desarrollo para disminuir los costes de desarrollo y mantenimiento

5.CONTENIDOS DE LA ASIGNATURA

UNIDAD DIDÁCTICA I

1. Introducción a la arquitectura del software: definición y relevancia.

Resumen: Cuando aumentan el tamaño y la complejidad del software, cuando las aplicaciones están formadas por multitud de componentes, la organización global del funcionamiento del sistema –la arquitectura del software— pasa a ser el aspecto más relevante.

Objetivos:

1. Definición de arquitectura de software.
2. El problema del diseño en software complejo y de altas prestaciones.
3. El punto de vista arquitectónico como solución para mejorar la comprensión de sistemas complejos.

2. Principios básicos del diseño arquitectónico.

Resumen: El arquitecto debe acondicionar el uso de las tecnologías de desarrollo para mejorar las expectativas de los objetivos de negocio.

Objetivos:

1. Tendencias en las tecnologías de desarrollo. Software procedimental, orientación a objetos y desarrollo con componentes.
2. Tecnologías propietarias, abiertas y comerciales. Uso de estándares.

UNIDAD DIDÁCTICA II

3. Estilos arquitectónicos.

Resumen: Una manera de ver una arquitectura es considerarla como un conjunto de componentes y la descripción de sus interacciones -los conectores-. Un estilo es un patrón arquitectónico aplicable a una familia de sistemas.

Objetivos:

1. Estilos arquitectónicos comunes.
2. Arquitecturas de componentes.
3. Arquitecturas de agentes.

4. Directrices arquitectónicas.

Resumen: ODP es un estándar formal que responde, definitivamente, a la pregunta de qué es la arquitectura de un



sistema.

Objetivos:

1. El modelo de negocio.
2. El modelo de la información.
3. El modelo computacional.
4. El modelo de ingeniería.
5. El modelo tecnológico.

5. Diseño arquitectónico. MDA

Resumen: El Desarrollo Dirigido por Modelos propugna que los elementos fundamentales en el desarrollo de software sean modelos abstractos de diseño en lugar de fragmentos de código. Su máximo exponente es la MDA (Model Driven Architecture) y persigue la reutilización de modelos traducibles a código.

Objetivos:

1. Entender los conflictos entre el comportamiento deseado para una aplicación y para la ejecución de su desarrollo y los aspectos técnicos de su implementación o la plataforma tecnológica utilizada para su desarrollo.
2. Estudiar la especificación de la arquitectura OMG MDA y comprender cómo independiza la descripción del comportamiento del software respecto a la tecnología utilizada para su implementación.

6. Diseño detallado. Patrones de diseño

Resumen: Los patrones son soluciones efectivas a problemas de diseño que pueden reutilizarse en nuevos problemas. Su uso se generalizó a partir de la publicación del libro de Erich Gamma et al. Patrones de Diseño.

Objetivos:

1. Analizar algunos problemas de diseño generalizados.
2. Comprender cómo un patrón de diseño resuelve un grupo numeroso de problemas de diseño.
3. Estudiar algunos problemas de diseño y el patrón solución correspondiente

6.EQUIPO DOCENTE

- [JOSE FELIX ESTIVARIZ LOPEZ](#)

7.METODOLOGÍA

La docencia de esta asignatura se impartirá a distancia, siguiendo el modelo educativo propio de la UNED adaptado al EEES. El principal instrumento docente será un curso virtual dentro de las plataformas educativas para la enseñanza a distancia, complementado con la asistencia personalizada del equipo docente y la tutela presencial y telemática.

Dentro del curso virtual el alumnado dispondrá de:

- Página de bienvenida, donde se indica el concepto general de la asignatura y se presenta el equipo docente.
- Calendario, donde se establece el orden temporal de actividades y sugerencias sobre el reparto temporal de la materia, para que el estudiante los adapte a su disponibilidad y necesidades.
- Materiales:
 - Guía didáctica del curso, donde se establecen los objetivos concretos y los puntos de interés.
 - Programa, donde se especifica la división del contenido por capítulos.
 - Procedimiento, donde se sugieren al alumno las tareas que debe realizar.
 - Ejemplos de trabajos, donde se orienta sobre las pruebas escritas y se muestran ejemplos de trabajos de cursos anteriores.
 - Pruebas de auto evaluación y de evaluación a distancia en línea.
- Comunicación:
 - Correo para comunicaciones individuales.
 - Foros de Debate donde se intercambian conocimientos y se resuelven dudas de tipo académico general.
 - Grupos de trabajo para intercambiar información dentro de los grupos.

Fuera del curso virtual el estudiante también tendrá acceso a realizar consultas al equipo docente a través del correo, teléfono y presencialmente en los horarios establecidos para estas actividades. También se podrán organizar



videoconferencias si las necesidades docentes lo hicieran preciso.

8.BIBLIOGRAFÍA BÁSICA

ISBN(13): 9780130274076
Título: SOFTWARE ARCHITECT BOOTCAMP
Autor/es: Mowbray, Thomas J. ;
Editorial: PRENTICE HALL

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9780201745726
Título: COMPONENT SOFTWARE - BEYOND OBJECT-ORIENTED PROGRAMMING (2)
Autor/es: Clemens Szyperski ;
Editorial: PEARSON EDUCATION

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9780201788914
Título: MDA DISTILLED
Autor/es: Stephen J. Miller ; Dirk Weise ; Axel Uhl ; Kendall Scott ;
Editorial: Addison-Wesley Professional. Addison-Wesley Object Technology Series

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9788420543673
Título: AGENTES SOFTWARE Y SISTEMAS MULTIAGENTE
Autor/es: Pérez, J. L. ; Pavón, J. ;
Editorial: PEARSON EDUCACIÓN

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED



Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

Comentarios y anexos:

DAVID GARLAN AND MARY SHAW. *An Introduction to Software Architecture*. 1994. Technical Report CMU/SEI-94-TR-021 http://www-2.cs.cmu.edu/afs/cs/project/able/ftp/intro_softarch/intro_softarch.pdf (Visitado en 26/03/2011)

BRUCE ECKEL. *Thinking in Patterns*. Libro electrónico. Versión 0.9, mayo de 2003. <http://www.pythoncriticalmass.com/http://www.mindviewinc.com/Books/downloads/TIPatterns-0.9.zip> (Visitado el 26/03/2011)

9.BIBLIOGRAFÍA COMPLEMENTARIA

Comentarios y anexos:

Model Driven Architecture. <http://www.omg.org/mda/>.

Erich Gamma, Kent Beck. *JUnit: A Cook's Tour*. Libro electrónico <http://junit.sourceforge.net/doc/cookstour/cookstour.htm>. (Visitado 26/03/2011)

10.RECURSOS DE APOYO AL ESTUDIO

La plataforma de e-Learning aLF, proporcionará el adecuado interfaz de interacción entre el alumno y sus profesores. aLF es una plataforma de e-Learning y colaboración que permite impartir y recibir formación, gestionar y compartir documentos, crear y participar en comunidades temáticas, así como realizar proyectos online.

Además, el equipo docente mantiene una [página Web con la asignatura](http://www.issi.uned.es/doctorado/softwarch/ArgDesSw.htm) en la que se mantienen contenidos, información y materiales en <http://www.issi.uned.es/doctorado/softwarch/ArgDesSw.htm>

Se ofrecerán las herramientas necesarias para que, tanto el equipo docente como el alumnado, encuentren la manera de compaginar tanto el trabajo individual como el aprendizaje cooperativo.

11.TUTORIZACIÓN Y SEGUIMIENTO

La tutorización de los alumnos se llevará a cabo fundamentalmente a través de los instrumentos de comunicación del curso virtual. También se atenderán consultas por teléfono y por correo electrónico por parte del equipo docente:

Horario:

Lunes de 16:00 a 20:00

Profesorado:

José Félix Estívariz López: Telf. 91-398.77.92

e-mail: softwarch@issi.uned.es

12.EVALUACIÓN DE LOS APRENDIZAJES

La calificación final de la asignatura se obtendrá a partir de los siguientes elementos:



- Trabajo de diseño arquitectónico del alumno, que se realizará a lo largo del semestre, con la supervisión y asesoramiento del equipo docente. Consistirá en una prospección, selección, documentación, instalación y análisis de varias herramientas de desarrollo orientado a MDA. Representará un 40 % de la calificación final.
- Trabajo de diseño con patrones del alumno, que se realizará a lo largo del semestre, con la supervisión y asesoramiento del equipo docente. Consistirá en el análisis y documentación de la aplicación de los patrones de diseño a un desarrollo a su alcance –propio o de su entorno profesional— o bien a librerías estándar de Java del tipo JUnit. Representará un 40 % de la calificación final.
- Participación en los foros y grupos de discusión del curso virtual en la plataforma aLF. Se evaluará el grado de participación, actitud constructiva y capacidad innovadora de las intervenciones y aportaciones. También se valorará la capacidad de expresión y la habilidad para argumentar ideas propias y aprovechar argumentos ajenos para deducir planteamientos nuevos. Representará un 20 % de la calificación final.

13.COLABORADORES DOCENTES

Véase equipo docente.

