



DESCRIPCIÓN DE LA ASIGNATURA

Grado/Máster en:	Master Universitario en INGENIERÍA INFORMÁTICA por la Universidad de Málaga
Centro:	Escuela Técnica Superior de Ingeniería Informática
Asignatura:	IMPLANTACIÓN DE METODOLOGÍAS DE DESARROLLO SOFTWARE
Código:	105
Tipo:	Obligatoria
Materia:	INGENIERÍA DEL SOFTWARE
Módulo:	TECNOLOGÍAS INFORMÁTICAS
Experimentalidad:	
Idioma en el que se imparte:	Español
Curso:	1
Semestre:	1
Nº Créditos	6
Nº Horas de dedicación del estudiante:	150
Nº Horas presenciales:	45
Tamaño del Grupo Grande:	
Tamaño del Grupo Reducido:	
Página web de la asignatura:	

EQUIPO DOCENTE

Departamento: LENGUAJES Y CIENCIAS DE LA COMPUTACIÓN

Área: LENGUAJES Y SISTEMAS INFORMÁTICOS

Nombre y Apellidos	Mail	Teléfono Laboral	Despacho	Horario Tutorías
Coordinador/a: EDUARDO FRANCISCO GUZMAN DE LOS RISCOS	eguzman@uma.es	952137146	3.2.27 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Viernes 09:15 - 13:15, Lunes 09:15 - 11:15 Segundo cuatrimestre: Viernes 09:30 - 13:30, Jueves 12:30 - 13:30, Martes 12:30 - 13:30

RECOMENDACIONES Y ORIENTACIONES

Se recomienda que esta asignatura se curse de forma simultánea a la de Gestión Integrada para el Desarrollo Software, puesto que ambas son complementarias. Asimismo también es recomendable que el alumnado posea a priori conocimientos básicos sobre Ingeniería del Software (metodologías, fases de desarrollo, etc.).

CONTEXTO

Esta asignatura se imparte en el primer semestre del máster. En ella se realizará un estudio avanzado y eminentemente empírico de las diferentes metodologías de desarrollo de aplicaciones software de media y gran envergadura, a través de proyectos sobre Internet de las Cosas, Computación en la Nube y Ciencia de Datos.

COMPETENCIAS

1 Competencias generales y básicas.

Competencias básicas

- 1.3 CB3. Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios
- 1.4 CB4. Que los estudiantes sepan comunicar sus conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin ambigüedades
- 1.5 CB5. Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

Competencias generales

- 1.5 CG5. Capacidad para la elaboración, planificación estratégica, dirección, coordinación y gestión técnica y económica de proyectos en todos los ámbitos de la Ingeniería en Informática siguiendo criterios de calidad y medioambientales.
- 1.6 CG6. Capacidad para la dirección general, dirección técnica y dirección de proyectos de investigación, desarrollo e innovación, en empresas y centros tecnológicos, en el ámbito de la Ingeniería Informática.
- 1.8 CG8. Capacidad para la aplicación de los conocimientos adquiridos y de resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinarios, siendo capaces de integrar estos conocimientos.
- 1.10 CG10. Capacidad para aplicar los principios de la economía y de la gestión de recursos humanos y proyectos, así como la legislación, regulación y normalización de la informática.

2 Competencias específicas.



2 Competencias específicas.

- 2.1 EDG1: Capacidad para la integración de tecnologías, aplicaciones, servicios y sistemas propios de la Ingeniería Informática, con carácter generalista, y en contextos más amplios y multidisciplinares.
- 2.4 ET11: Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas, servicios y contenidos informáticos.
- 2.8 ET15: Capacidad para analizar las necesidades de información que se plantean en un entorno y llevar a cabo en todas sus etapas el proceso de construcción de un sistema de información.
- 2.14 ET111. Capacidad para conceptualizar, diseñar, desarrollar y evaluar la interacción persona- ordenador de productos, sistemas, aplicaciones y servicios informáticos.

3 Competencias transversales.

- 3.2 CT2. Capacidad para identificar estrategias, herramientas y métodos que responden a situaciones de éxito que pueden ser abordadas con los recursos disponibles.

CONTENIDOS DE LA ASIGNATURA

Metodologías

1. El ciclo de vida y las metodologías de desarrollo.
2. Adecuación de las metodologías a cada problema.
3. Desarrollo en espiral y en cascada.
4. Proceso Unificado.
5. Desarrollo iterativo e incremental.
6. Metodologías ágiles.
7. Desarrollo Lean.
8. Modelos de procesos.
9. Casos de éxito y fracaso.

Tecnologías

- Internet de las Cosas (IoT)
- Computación en la Nube (Cloud Computing)
- Ingeniería y Ciencia de Datos

ACTIVIDADES FORMATIVAS

Actividades presenciales

Actividades expositivas

- Lección magistral

Actividades no presenciales

Actividades de elaboración de documentos

- Elaboración de informes

Actividades prácticas

- Desarrollo y evaluación de proyectos

ACTIVIDADES DE EVALUACIÓN

Actividades de evaluación presenciales

Actividades de evaluación del estudiante

- Autoevaluación del estudiante
- Coevaluación
- Realización de trabajos y/o proyectos
- Participación en clase

RESULTADOS DE APRENDIZAJE / CRITERIOS DE EVALUACIÓN

El objetivo es que el alumnado, a través de un proceso de aprendizaje significativo, sea capaz de determinar cuál es la metodología más adecuada para cada proyecto software, evaluar la idoneidad de aplicar una determinada metodología para un entorno de desarrollo, y conocer los fundamentos



básicos para el desarrollo de proyectos sobre IoT, Cloud Computing y Ciencia de Datos. Igualmente el alumnado deberá ser capaz de analizar y posteriormente diagnosticar los errores cometidos por un equipo durante el proceso de desarrollo, proponiendo la metodología más adecuada en cada caso y las pautas a seguir para la mejora del funcionamiento del equipo. Más concretamente los objetivos de aprendizaje principales que se persiguen con esta asignatura son los siguientes:

- Ser capaz de implantar aplicaciones en la nube usando plataformas y frameworks de desarrollo.
- Diseñar y desarrollar software para interactuar con sensores, actuadores y dispositivos empujados diversos que funcionen en condiciones heterogéneas.
- Conocer plataformas para facilitar el desarrollo de aplicaciones IoT
- Ser capaz de desarrollar un proyecto de Ingeniería de Datos siguiendo una metodología adecuada.

PROCEDIMIENTO DE EVALUACIÓN

Al aplicarse como metodología pedagógica el aprendizaje basado en proyectos, el proceso de evaluación será continuo. Por cada proyecto se evaluará el trabajo realizado por el equipo de forma global, así como la participación individual en el mismo. Para esto último se hará uso de la autoevaluación y de la evaluación por pares. Los aspectos a evaluar serán principalmente los siguientes: conocimiento de las metodologías de desarrollo, conocimiento y uso de las tecnologías específicas de cada proyecto (IoT, cloud computing y ciencia de datos), capacidad para trabajar en grupo, participación activa en la toma de decisiones, involucración en el grupo, etc.

Adicionalmente se realizará una prueba de tipo test (o preguntas cortas) para evaluar el grado de asimilación de los conocimientos sobre el bloque de contenido sobre tecnologías.

Se valorará también la asistencia a clase y la puntualidad.

- Trabajo en grupo: 80%
- Asistencia a clase y puntualidad: 10%.
- Examen escrito: 10%.

Para el resto de convocatorias, la evaluación consistirá en dos partes:

Un examen teórico sobre los contenidos trabajados durante la asignatura (que representará el 60% de la nota final), y la elaboración de un trabajo práctico individual que el alumno deberá defender (40%).

Para la realización del trabajo el alumno deberá ponerse en contacto con el profesorado de la asignatura al comienzo del periodo de exámenes correspondiente.

BIBLIOGRAFÍA Y OTROS RECURSOS

Básica

- Brooks, F.P. (1995). The Mythical Man Month: Essays on Software Engineering. Prentice-Hall.
- Cohn, Mike. (2005). Agile Estimating and Planning. Prentice Hall.
- DeMarco, T. y Lister, T. (1999). Peopleware: Productive Projects and Teams. Dorset House Publishing Company.
- Fowler, M. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison Wesley.
- Martin, R.C. (2002). Agile Software Development, Principles, Patterns, and Practices. Prentice Hall.
- McConnell, S. (1996). Code Complete: A Practical Handbook of Software Construction. Microsoft Press.
- McConnell, S. (1996). Rapid Development: Taming Wild Software Schedules. Microsoft Press.
- McConnell, S. (2006). Software Estimation: Demystifying the Black Art. Microsoft Press.
- Poppendieck, M. y Poppendieck, T. (2003). Lean Software Development: An Agile Toolkit. Addison-Wesley Professional.
- Rasmusson, J. (2010). The Agile Samurai: How Agile Masters Deliver Great Software. The Pragmatic Bookshelf.
- Schwaber, K. (2004). Agile Project Management with Scrum. Microsoft Press.

DISTRIBUCIÓN DEL TRABAJO DEL ESTUDIANTE

ACTIVIDAD FORMATIVA PRESENCIAL

Descripción	Horas	Grupo grande	Grupos reducidos
Lección magistral	45	<input checked="" type="checkbox"/>	<input type="checkbox"/>

TOTAL HORAS ACTIVIDAD FORMATIVA PRESENCIAL 45

ACTIVIDAD FORMATIVA NO PRESENCIAL

Descripción	Horas
Desarrollo y evaluación de proyectos	80
Elaboración de informes	10

TOTAL HORAS ACTIVIDAD FORMATIVA NO PRESENCIAL 90

TOTAL HORAS ACTIVIDAD EVALUACIÓN 15

TOTAL HORAS DE TRABAJO DEL ESTUDIANTE 150

