



DESCRIPCIÓN DE LA ASIGNATURA

Grado/Máster en:	Graduado/a en Ingeniería del Software por la Universidad de Málaga
Centro:	Escuela Técnica Superior de Ingeniería Informática
Asignatura:	Desarrollo de Software Crítico
Código:	803
Tipo:	Optativa
Materia:	Complementos de Sistemas Distribuidos
Módulo:	Complementos de la Ingeniería Informática
Experimentalidad:	69 % teórica y 31 % práctica
Idioma en el que se imparte:	Español
Curso:	4
Semestre:	1
Nº Créditos	6
Nº Horas de dedicación del estudiante:	150
Nº Horas presenciales:	60
Tamaño del Grupo Grande:	72
Tamaño del Grupo Reducido:	30
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: MANUEL DIAZ RODRIGUEZ	mdiaz@uma.es	952131394	3.2.13 - E.T.S.I. INFORMÁTICA	Primer cuatrimestre: Lunes 08:30 - 11:30, Viernes 08:30 - 11:30
DANIEL GARRIDO MARQUEZ	dgm@uma.es	952131401	3.2.1 - E.T.S.I. INFORMÁTICA	Todo el curso: Martes 09:00 - 10:30, Miércoles 09:00 - 12:30, Martes 12:30 - 13:30

RECOMENDACIONES Y ORIENTACIONES

El alumno debe conocer en detalle las distintas fases de desarrollo software, especialmente las prácticas asociadas a las fases de verificación y validación de software, desde el punto de vista del proceso y de las tecnologías asociadas. También debe entender la problemática del desarrollo de sistemas empotrados, especialmente desde el punto de vista de las pruebas y mantenimiento y los aspectos de tiempo real y concurrencia.

CONTEXTO

El papel del software en los sistemas críticos es cada día más importante. Por otro lado, las técnicas de desarrollo software evolucionan constantemente y esta evolución no puede ser seguida fácilmente en el desarrollo de los sistemas críticos. La asignatura analiza los sistemas de desarrollo de software crítico actuales, considerando su evolución y el impacto de las nuevas tecnologías hardware y software en el corto y medio plazo.

La asignatura cubre todas las fases del desarrollo, haciendo especial hincapie en los aspectos relacionados con los requisitos y la validación.

COMPETENCIAS

1 Competencias generales y básicas.

GENERALES

- CG04** Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias básicas, comunes y específicas del título.
- CG08** Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.
- CG09** Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.
- CG10** Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias básicas, comunes y específicas del título.
- CG13** Capacidad de expresión oral y escrita en un segundo idioma (inglés: acreditación por entidad homologada del nivel B1).

2 Competencias específicas.

Formación común

- CC08** Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.



2 Competencias específicas.

Formación común

- CC11** Conocimiento y aplicación de las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores e Internet y diseñar e implementar aplicaciones basadas en ellas.
- CC13** Conocimiento y aplicación de las herramientas necesarias para el almacenamiento, procesamiento y acceso a los Sistemas de información, incluidos los basados en web.
- CC14** Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.
- CC16** Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.

CONTENIDOS DE LA ASIGNATURA

Introducción a los Sistemas Críticos

- 1.1. Características. El papel del software
- 1.2. Historia y evolución. Tecnologías y accidentes
- 1.3. Terminología
- 1.4. Estándares y certificación

Análisis de Sistemas y Software Crítico

- 2.1. Análisis de peligros y riesgos
- 2.2.. Estudio de la seguridad
- 2.3. Análisis de fallos
- 2.4. Relación entre los distintos tipos de análisis. Ejemplos
- 2.5. Componentes certificados y no certificados: COTS y SOUP

Patrones de Diseño

- 3.1. Balances arquitectónicos: disponibilidad/fiabilidad, seguridad/prestaciones, ¿
- 3.2. Patrones para gestión de errores
- 3.3. Replicación y diversificación

Validación del Diseño

- 4.1. Modelos de Markov
- 4.2. Arbol de fallos. FTA y FMECA
- 4.3. Verificación de diseño semi-formal y formal

Codificación

- 5.1. Guías de codificación en los estándares
- 5.2. SPARK Ada 2014
- 5.3. Métricas de cobertura de código
- 5.4. Análisis estático

Inyección de fallos Back-to-back testing Validación de la ccadena de herramientas Verificación del compilador

- 6.1. Inyección de fallos
- 6.2. Back-to-back testing
- 6.3. Validación de la ccadena de herramientas
- 6.4. Verificación del compilador

ACTIVIDADES FORMATIVAS

Actividades presenciales

Actividades expositivas

Lección magistral

Actividades prácticas en instalaciones específicas

Prácticas en laboratorio

ACTIVIDADES DE EVALUACIÓN

Actividades de evaluación presenciales



Actividades de evaluación presenciales

Actividades de evaluación del estudiante

- Examen parcial
- Examen final
- Realización de trabajos y/o proyectos

RESULTADOS DE APRENDIZAJE / CRITERIOS DE EVALUACIÓN

Los alumnos deberán:

- Conocer los principios relacionados con la seguridad del software y los distintos niveles de criticidad y tipos de fallos
- Conocer las técnicas básicas de análisis de fallos (FMA y FTA)
- Analizar los requisitos relacionados con la seguridad software y diseñar arquitecturas adecuadas para los distintos niveles de criticidad
- Diseñar e implementar sistemas tolerantes a fallos, tanto software como hardware
- Conocer las distintas aproximaciones al desarrollo de software crítico en el campo de los sistemas empujados, con especial énfasis en los dominios de automoción, aviónica y nuclear

Se evaluará la correcta adquisición de estos resultados de aprendizaje mediante pruebas parciales, entrega de prácticas y los exámenes que se realizarán en las convocatorias oficiales. En las pruebas y exámenes se incluirán tanto los conceptos teóricos de la asignatura (CG08) como problemas del área del estudio de la misma que puedan resultar novedosos para los estudiantes (CB02 y CG09). Para la resolución de dichos problemas, será necesario que el estudiante demuestre cierto grado de autonomía (CB05). Se evaluará no sólo la corrección técnica de las soluciones propuestas, sino también su correcta redacción utilizando la terminología y estilo adecuados (CB04 y CB02).

PROCEDIMIENTO DE EVALUACIÓN

La evaluación de la asignatura se hará de la siguiente manera (se incluye el peso de cada componente en la calificación final):

- (A) AEP 1.4 Evaluación continua y examen parcial a mediados del cuatrimestre (60%)
- (B) AEP 1.6 Realización de trabajos y/o proyectos (40%)

AEP 1.5 Examen final. Los alumnos tendrán que presentarse únicamente si no han superado alguno de los apartados anteriores. El examen incluirá un apartado para cada componente de la evaluación (apartados A y B) y con el mismo peso indicado (60 y 40%)

Para aprobar la asignatura se requerirá una nota mínima de 4/10 en cada uno de los dos componentes de la calificación, tanto si se aprueba por evaluación continua, como en el examen final.

Las calificaciones de los parciales y los trabajos se guardan para el examen de septiembre. Tanto en esta segunda convocatoria como en las extraordinarias se incluirá la posibilidad de examinarse del 100% de la asignatura.

Los alumnos con reconocimiento de estudiantes de tiempo parcial o deportistas de alto nivel podrán acordar con el profesor la posibilidad de realizar las pruebas parciales en otras fechas en el caso de que fuera imposible su asistencia, siempre con previo aviso y justificación.

BIBLIOGRAFÍA Y OTROS RECURSOS

Básica

- Building High Integrity Applications with SPARK; J.W. McCormick, Peter C. Chapin; Cambridge University Press; 2015
- Concurrent and Real-Time Programming in Ada; Alan Burns, Andy Wellings; Cambridge University Press; 2007
- Embedded Software Development for Safety-Critical Systems; Chris Hobbs; CRC Press; 2016
- Engineering a Safer World, Systems Thinking Applied to Safety; MIT Press; Nancy G. Leveson; 2011
- Programming in Ada 2012; John Barnes; Cambridge University Press; 2014
- Safety Critical Computer Systems; Neil Storey; Addison-Wesley; 1996
- Safety critical systems handbook [Recurso electrónico]: a straightforward guide to functional safety : IEC 61508 (2010 Edition) and related standards : including process IEC 61511, Machinery IEC 62061 and ISO 13849 / David J. Smith, Kenneth Gl. Simpson
- SAFWARE, System Safety and Computers; Nancy G. Leveson; Addison-Wesley; 1995

DISTRIBUCIÓN DEL TRABAJO DEL ESTUDIANTE

ACTIVIDAD FORMATIVA PRESENCIAL

Descripción	Horas	Grupo grande	Grupos reducidos
Lección magistral	41,4	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Prácticas en laboratorio	18,6	<input type="checkbox"/>	<input checked="" type="checkbox"/>

TOTAL HORAS ACTIVIDAD FORMATIVA PRESENCIAL 60

TOTAL HORAS ACTIVIDAD FORMATIVA NO PRESENCIAL 75



TOTAL HORAS ACTIVIDAD EVALUACIÓN

Horas

Grupo grande Grupos reducidos

TOTAL HORAS DE TRABAJO DEL ESTUDIANTE

150

