

Facultad de Ciencias

GUÍA DOCENTE DE LA ASIGNATURA

M1692 - Desarrollo de Software para Sistemas Empotrados

Máster Universitario en Ingeniería Informática
Obligatoria. Curso 1

Curso Académico 2021-2022

1. DATOS IDENTIFICATIVOS

Título/s	Máster Universitario en Ingeniería Informática	Tipología v Curso	Obligatoria. Curso 1
Centro	Facultad de Ciencias		
Módulo / materia	INGENIERÍA DEL SOFTWARE TECNOLOGÍAS INFORMÁTICAS		
Código y denominación	M1692 - Desarrollo de Software para Sistemas Empotrados		
Créditos ECTS	6	Cuatrimestre	Cuatrimestral (1)
Web	http://www.istr.unican.es/ asignaturas/dsw_empotrados		
Idioma de impartición	Español	English friendly	Sí
		Forma de impartición	Presencial

Departamento	DPTO. INGENIERÍA INFORMÁTICA Y ELECTRÓNICA
Profesor responsable	MICHAEL GONZALEZ HARBOUR
E-mail	michael.gonzalez@unican.es
Número despacho	Facultad de Ciencias. Planta: + 3. DESPACHO PROFESORES (3055)
Otros profesores	HECTOR PEREZ TIJERO

2. CONOCIMIENTOS PREVIOS

Se requieren conocimientos de programación C, sistemas de tiempo real, sistemas operativos, redes de computadores, ingeniería de software, y programación concurrente, de nivel equivalente al obtenido en las asignaturas obligatorias del Grado de Informática de la Universidad de Cantabria

3. COMPETENCIAS GENÉRICAS Y ESPECÍFICAS DEL PLAN DE ESTUDIOS TRABAJADAS

Competencias Genéricas
Capacidad para proyectar, calcular y diseñar productos, procesos e instalaciones en todos los ámbitos de la Ingeniería Informática
Capacidad para la dirección de obras e instalaciones de sistemas informáticos, cumpliendo la normativa vigente y asegurando la calidad del servicio
Capacidad para el modelado matemático, cálculo y simulación en centros tecnológicos y de ingeniería de empresa, particularmente en tareas de investigación, desarrollo e innovación en todos los ámbitos relacionados con la Ingeniería Informática
Capacidad para la aplicación de los conocimientos adquiridos y de resolver problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares, siendo capaces de integrar estos conocimientos
Competencias Específicas
Capacidad para asegurar, gestionar, auditar y certificar la calidad de los desarrollos, procesos, sistemas, servicios, aplicaciones y productos informáticos
Capacidad para diseñar y desarrollar sistemas, aplicaciones y servicios informáticos en sistemas empotrados y ubicuos
Competencias Básicas
Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación
Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio
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
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
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 Transversales
Capacidad de análisis, síntesis y evaluación
Capacidad de organización y planificación
Capacidad de resolución de problemas aplicando técnicas de ingeniería
Capacidad de razonamiento crítico
Aprendizaje autónomo
Creatividad
Capacidad de iniciativa y espíritu emprendedor

3.1 RESULTADOS DE APRENDIZAJE

- Conocer y aplicar las técnicas y conceptos distintivos del desarrollo de software para sistemas empotrados, en los que el sistema mantiene una fuerte relación reactiva con su entorno físico de operación
 - Identificar las limitaciones así como los requisitos funcionales y extra-funcionales a validar que diferencian los sistemas empotrados de los sistemas informáticos de propósito general
 - Saber plantear el desarrollo de software para sistemas empotrados atendiendo a la forma de abordar la interacción con su entorno real de operación, sea ésta conducida por eventos o por tiempo
 - Conocer metodologías, técnicas, patrones y estándares útiles para el desarrollo de software para sistemas empotrados, tanto a nivel de la definición arquitectural como de su implementación y validación

4. OBJETIVOS

Alcanzar los resultados del aprendizaje expuestos

5. MODALIDADES ORGANIZATIVAS Y MÉTODOS DOCENTES

ACTIVIDADES	HORAS DE LA ASIGNATURA
ACTIVIDADES PRESENCIALES	
HORAS DE CLASE (A)	
- Teoría (TE)	20
- Prácticas en Aula (PA)	10
- Prácticas de Laboratorio Experimental(PLE)	30
- Prácticas de Laboratorio en Ordenador (PLO)	
- Prácticas Clínicas (CL)	
Subtotal horas de clase	60
ACTIVIDADES DE SEGUIMIENTO (B)	
- Tutorías (TU)	7,5
- Evaluación (EV)	7,5
Subtotal actividades de seguimiento	15
Total actividades presenciales (A+B)	75
ACTIVIDADES NO PRESENCIALES	
Trabajo en grupo (TG)	5
Trabajo autónomo (TA)	70
Tutorías No Presenciales (TU-NP)	
Evaluación No Presencial (EV-NP)	
Total actividades no presenciales	75
HORAS TOTALES	150

6. ORGANIZACIÓN DOCENTE													
CONTENIDOS		TE	PA	PLE	PLO	CL	TU	EV	TG	TA	TU- NP	EV- NP	Semana
1	1. Introducción. Sistemas empotrados. Planificación de las aplicaciones software en sistemas reactivos: dirigida por el tiempo y dirigida por eventos. Modelos para planificación de recursos que no son de procesamiento: energía, redes de comunicación y memoria. Variaciones al proceso de desarrollo. El papel del desarrollo de software dirigido por modelos.	6,00	0,00	0,00	0,00	0,00	1,00	0,00	1,00	12,00	0,00	0,00	1-3
2	2. Plataformas para sistemas empotrados. Funcionamiento sobre máquina desnuda: ejecutivos cíclicos. Sistemas operativos dirigidos por eventos. Reserva de recursos. Particionado en el espacio y el tiempo. Particionamiento hardware-software. Plataformas basadas en lenguajes de programación concurrentes. Sistemas empotrados distribuidos.	4,00	0,00	6,00	0,00	0,00	1,50	0,50	1,00	12,00	0,00	0,00	4-6
3	3. Especificación y análisis de requisitos software en sistemas empotrados. Introducción. Especificación de requisitos en sistemas reactivos. UML/MARTE. Use Case Maps. RDAL/AADL.	2,00	2,00	6,00	0,00	0,00	1,50	0,50	1,00	12,00	0,00	0,00	7,8
4	4. Diseño arquitectónico en sistemas empotrados. Introducción. Diseño basado en modelos de control. Diseño basado en componentes. Lenguajes de descripción de arquitecturas. AADL. Análisis y generación de código Patrones arquitecturales. Patrones para tiempo real.	4,00	3,00	8,00	0,00	0,00	1,50	0,50	1,00	12,00	0,00	0,00	9-11
5	5. Implementación software de sistemas empotrados. Instalación y configuración del entorno de desarrollo. Uso básico de dispositivos de entrada/salida. Caracterización de la plataforma. Generación automática de código. Proyecto de un sistema empotrado.	4,00	5,00	10,00	0,00	0,00	2,00	1,00	1,00	22,00	0,00	0,00	12,13
6	Evaluación final	0,00	0,00	0,00	0,00	0,00	0,00	5,00	0,00	0,00	0,00	0,00	16
TOTAL DE HORAS		20,00	10,00	30,00	0,00	0,00	7,50	7,50	5,00	70,00	0,00	0,00	

Esta organización tiene carácter orientativo.

Ante la situación incierta de que las medidas de distanciamiento social establecidas por las autoridades sanitarias no permitan desarrollar alguna actividad docente de forma presencial en el aula para todos los estudiantes matriculados, se adoptará una modalidad mixta de docencia que combine esta docencia presencial en el aula con docencia a distancia. De la misma manera, la tutorización podrá ser sustituida por tutorización a distancia utilizando medios telemáticos.

TE	Horas de teoría
PA	Horas de prácticas en aula
PLE	Horas de prácticas de laboratorio experimental
PLO	Horas de prácticas de laboratorio en ordenador
CL	Horas de prácticas clínicas
TU	Horas de tutoría
EV	Horas de evaluación
TG	Horas de trabajo en grupo
TA	Horas de trabajo autónomo
TU-NP	Tutorías No Presenciales
EV-NP	Evaluación No Presencial

7. MÉTODOS DE LA EVALUACIÓN

Descripción	Tipología	Eval. Final	Recuper.	%
Clases de problemas	Trabajo	No	Sí	20,00
Calif. mínima	0,00			
Duración	Durante el curso			
Fecha realización	Evaluación continua			
Condiciones recuperación	Entrega de los ejercicios antes del final del curso			
Observaciones	Evaluación de los trabajos presentados durante las clases de prácticas en aula (problemas)			
Prácticas	Evaluación en laboratorio	No	Sí	30,00
Calif. mínima	4,00			
Duración	Durante el curso			
Fecha realización	Evaluación continua			
Condiciones recuperación	Entrega de las prácticas antes del final del curso			
Observaciones	Evaluación de las prácticas, evaluando el trabajo en el laboratorio, y los informes de las prácticas. Se penalizará la entrega de los informes pasado el plazo que se establezca.			
Examen Final	Examen escrito	Sí	Sí	50,00
Calif. mínima	4,00			
Duración	4 horas			
Fecha realización	Periodo de exámenes de Febrero			
Condiciones recuperación	Examen en septiembre			
Observaciones	Es un examen en ordenador sobre los aspectos vistos durante las clases de prácticas en aula. Se podrán llevar apuntes y libros.			
TOTAL				100,00
Observaciones				
Hay una única convocatoria anual. Si la asignatura no se supera en las actividades de evaluación ordinarias realizadas en el primer cuatrimestre o en el segundo se podrá acceder a la evaluación de recuperación en septiembre.				
Si el cupo de matrículas de honor de la asignatura se completa en la evaluación ordinaria, los alumnos que se presenten a la recuperación no podrán optar a la calificación de matrícula de honor.				
La media ponderada de problemas y prácticas tiene una nota mínima de 4.0.				
Criterios de evaluación para estudiantes a tiempo parcial				
La evaluación continua de problemas y prácticas se podrá superar por los estudiantes a tiempo parcial entregando los trabajos antes del final del curso.				

8. BIBLIOGRAFÍA Y MATERIALES DIDÁCTICOS

BÁSICA

"Embedded System Design: Embedded Systems Foundations of Cyber-Physical Systems". Peter Marwedel. Springer, 2010

"Real-Time Systems: Design Principles for Distributed Embedded Applications". Hermann Kopetz. Springer, 2011

Complementaria

“Software Engineering for Embedded Systems: Methods, Practical Techniques, and Applications”. Robert Oshana, Mark Kraeling. Newnes, 2013

“Design Patterns for Embedded Systems in C: An Embedded Software Engineering Toolkit”. Bruce Powel Douglass. Elsevier, 2010

“Model-Based Engineering of Embedded Real-Time Systems”: International Dagstuhl Workshop, Dagstuhl Castle, Germany, November 4-9, 2007. Revised Selected Papers. Holger Giese, Gabor Karsai, Edward A. Lee, Bernhard Rumpe, Bernhard Schätz. Springer, 2010

“Embedded Systems: Analysis and Modeling with SysML, UML and AADL”. Fabrice Kordon, Jérôme Hugues, Agustí Canals, Alain Dohet. John Wiley & Sons, 2013

“Modeling and Analysis of Real-Time and Embedded Systems with UML and MARTE”. Bran Selic and Sébastien Gérard. Elsevier 2014.

9. SOFTWARE

PROGRAMA / APLICACIÓN	CENTRO	PLANTA	SALA	HORARIO
Entorno de programación Eclipse versión 2021 o posterior	Facultad de Ciencias	1	Laboratorio de sistemas de tiempo real	

10. COMPETENCIAS LINGÜÍSTICAS

- Comprensión escrita Comprensión oral
 Expresión escrita Expresión oral
 Asignatura íntegramente desarrollada en inglés

Observaciones

Lectura de documentación técnica