

Guía docente de la asignatura

Desarrollo de Software para Sistemas Empotrados

Fecha última actualización: 07/07/2021
 Fecha de aprobación por la Comisión Académica: 26/07/2021

Máster

Máster Universitario en Desarrollo del Software

MÓDULO

Módulo 6: Ingeniería del Software de Sistemas Distribuidos, Empotrados y de Tiempo Real

RAMA

Ingeniería y Arquitectura

CENTRO RESPONSABLE DEL TÍTULO

Escuela Internacional de Posgrado

Semestre

Primero

Créditos

3

Tipo

Optativa

Tipo de enseñanza

Presencial

PRERREQUISITOS Y/O RECOMENDACIONES

No hay prerrequisitos ni recomendaciones.

BREVE DESCRIPCIÓN DE CONTENIDOS (Según memoria de verificación del Máster)

- Introducción a los sistemas empotrados. Características principales. Particionamiento hardware-software. Hardware en sistemas empotrados.
- Diseño de un sistema empotrado. Lenguajes de programación. Entornos de ejecución: firmware, núcleos y sistemas operativos. Despliegue de una aplicación: esquema host-target.
- Tiempo real en Sistemas Empotrados. Gestión y control del tiempo. Planificación de tareas de tiempo real.
- Métodos y técnicas avanzadas para el desarrollo de proyectos con aplicaciones empotradas. Desarrollo de software dirigido por modelos.
- Aplicaciones actuales y nuevas vías de investigación. Redes de sensores, automatización industrial, inteligencia ambiental, instrumentación virtual, control distribuido, internet de las cosas, sensores inteligentes, mecatrónica,...



- Introduction to embedded systems. Main features. The Hardware-Software Partition. Hardware in embedded systems.
- Design of an embedded system. Programming Languages. Runtime environments: firmware, kernels and operating systems. Deployment of an application: host-target scheme.
- Real Time in embedded systems. Management and control of time. Scheduling of real-time tasks.
- Methods and advanced techniques for the development of projects with embedded applications. Development of model-driven software.
- Current applications and new lines of research. Sensor networks, industrial automation, ambient intelligence, virtual instrumentation, distributed control, smart sensors, mechatronics, ...

COMPETENCIAS

COMPETENCIAS BÁSICAS

- CB6 - Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación.
- CB7 - 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.
- CB8 - 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.
- CB9 - 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.
- CB10 - 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

- CG01 - Habilidades cognitivas: conocer los principales problemas o retos tecnológicos planteados en el ámbito de las líneas de investigación del programa de posgrado, conocer los principios de las técnicas o metodologías de solución para dichos problemas propuestas por la comunidad científica, conocer las debilidades y fortalezas de dichas soluciones, así como conocer las aplicaciones que este conocimiento tiene en la sociedad actual.
- CG02 - Destreza para iniciar un trabajo de investigación científica o desarrollo tecnológico original e innovador, en el marco de los problemas descritos en el punto anterior.
- CG03 - Ser capaz de emplear el conocimiento científico existente en la resolución de problemas o mejora de procesos a nivel individual o en el contexto de empresas u organismos públicos.



- CG04 - Capacidades sistémicas para obtener la capacidad de asimilación y adaptación a la evolución futura del estado del arte en el ámbito de las disciplinas científicas del Máster.
- CG05 - Destrezas tecnológicas: capacidad de usar, evaluar, crear, modificar o extender la herramientas informáticas útiles en la resolución de problemas relacionados con las líneas de investigación
- CG06 - Capacidades metodológicas: conocer las principales fuentes bibliográficas que describen los avances científicos en las líneas de investigación del programa de posgrado.
- CG07 - Destrezas lingüísticas: conocer y utilizar la terminología científica especializada, tanto en español como en inglés, relacionada con las líneas de investigación del departamento.
- CG08 - Competencias personales: capacidad de análisis y síntesis en la resolución efectiva de problemas, así como capacidad de toma de decisiones, organización y planificación. Capacidad de comunicación escrita y oral.
- CG09 - Competencias interpersonales: capacidad de trabajo en equipo, incluyendo la toma de decisiones en colectivos o grupos. Habilidades en las relaciones interpersonales. Habilidades para presentar trabajos y mantener debates en grupo.
- CG10 - Destrezas de redacción: ser capaz de expresar los resultados y el desarrollo de las investigaciones en textos o informes científico-técnicos, conocer los mecanismos de revisión entre pares propios de la ciencia para estos documentos, así como los mecanismos para su difusión en forma de artículos en revistas, libros, sitios web o en aportaciones a congresos.

COMPETENCIAS ESPECÍFICAS

- CE01 - Ser capaz de llevar a cabo un trabajo de investigación en campos científicos relacionados con el desarrollo del software, teniendo en cuenta los recursos disponibles y sus implicaciones éticas y sociales
- CE03 - Identificar y comprender los conceptos clave y las principales características de los sistemas software, en cuanto a sus requerimientos, diseño o programación, así como saber aplicar los principales modelos, métodos y técnicas de la Ingeniería del Software al desarrollo de estos sistemas.
- CE04 - Conocer y saber aplicar métodos, técnicas y herramientas avanzadas de modelado, análisis, diseño y simulación en sistemas colaborativos, ubicuos, móviles, distribuidos, de diálogo, empotrados, de tiempo real o de procesos de negocio.
- CE05 - Identificar y valorar propiedades software de usabilidad, accesibilidad, seguridad, confiabilidad, rendimiento, y ética informática, entre otras, y analizar cómo afectan a la calidad de un sistema software.
- CE06 - Saber aplicar las estrategias de modelado más adecuadas para el diseño de sistemas software, así como las técnicas para la generación sistemática de sistemas dirigido por modelos
- CE07 - Diseñar y desarrollar sistemas software desde una perspectiva centrada en el usuario.
- CE08 - Diseñar modelos de sistemas software que permitan aplicar mecanismos evolutivos de reflexión, parametrización, refactorización, reutilización y simulación de procesos, entre otros.
- CE09 - Conocer los paradigmas, fundamentos y técnicas específicas de interacción persona-ordenador para el diseño de sistemas software de interacción multimodales (voz, tangibles, gestos)
- CE10 - Comprender las metodologías y técnicas asociadas al desarrollo e implantación de aplicaciones web, en cuanto al sistema hipermedia construido, al gestor de contenido seleccionado, o la tecnología de desarrollo web utilizada en su implementación, así como comprender las diferencias existentes en cuanto al diseño y desarrollo frente a otros tipos



de aplicaciones.

- CE11 - Reconocer y analizar los métodos y técnicas de sistemas de acceso integrado a múltiples fuentes de datos, en cuanto a los modelos espaciales y temporales para el diseño de base de datos, los almacenes de datos y sistemas OLAP, las ontologías y/o la web semántica.
- CE12 - Comprender y conocer técnicas de representación, interconexión, implementación, despliegue, y reutilización de servicios y componentes software y de negocio para su aplicación en sistemas colaborativos, distribuidos, ubicuos, empotrados y/o de tiempo real.
- CE15 - Conocer y comprender los fundamentos, técnicas y herramientas básicas para la programación eficiente de algoritmos gráficos y en particular en arquitecturas de altas prestaciones como las unidades de procesamiento de gráficos (GPU).

COMPETENCIAS TRANSVERSALES

- CT01 - Mostrar interés por la calidad y la excelencia en la realización de diferentes tareas.
- CT02 - Comprender y defender la importancia que la diversidad de culturas y costumbres tienen en la investigación o práctica profesional.
- CT03 - Tener un compromiso ético y social en la aplicación de los conocimientos adquiridos.
- CT04 - Ser capaz de trabajar en equipos interdisciplinarios para alcanzar objetivos comunes desde campos expertos diferenciados.
- CT05 - Incorporar los principios del Diseño Universal en el desempeño de su profesión.

RESULTADOS DE APRENDIZAJE (Objetivos)

El alumno sabrá/comprenderá:

- Reconocer las particularidades especiales de los sistemas empotrados en cuanto a su requerimiento, diseño o programación.
- Saber abordar las cuestiones de diseño de sistemas empotrados en sus variados ámbitos software-hardware.
- Conocer las técnicas de programación, lenguajes y herramientas más utilizadas para la programación de este tipo de sistemas sobre sistemas operativos convencionales o de tiempo real.
- Evaluar metodologías de diseño de desarrollo de software para entornos empotrados.

El alumno será capaz de:

- Diseñar y desplegar una aplicación empotrada sobre un target concreto atendiendo a los requerimientos específicos de la aplicación.
- Aplicar una metodología o proceso de desarrollo para el desarrollo de software para un entorno empotrado.
- Implementar aplicaciones sobre sistemas empotrados sin entornos de ejecución y con entornos



de ejecución como, por ejemplo, un sistema operativo de tiempo real.

- Realizar pequeños proyectos de sistemas empotrados para casos prácticos realistas de domótica, telecomunicaciones, instrumentación, manufactura,... según los intereses del alumno y su orientación en el Máster.

El alumno tendrá actitudes para:

- Trabajar en equipo asumiendo el rol que corresponde dentro de un proyecto.
- Manifestar interés por los distintos procesos de innovación ligados a las TICs, nuevas técnicas y procedimientos.
- Valorar el impacto social que provocan los sistemas digitales en la sociedad, y en particular los sistemas empotrados.
- Favorecer la creatividad y el razonamiento crítico.
- Comunicación oral y escrita.
- Tener un compromiso ético ante el software y su desarrollo.

PROGRAMA DE CONTENIDOS TEÓRICOS Y PRÁCTICOS

TEÓRICO

1. Introducción a los sistemas empotrados. Consideraciones y requerimientos particulares del diseño de sistemas empotrados: criticidad, restricciones de tiempo real, coste por unidad, consumo de energía,... Hardware básico de un sistema empotrado. Comparación entre distintas arquitecturas de plataformas empotradas. Software básico de un sistema empotrado. Desarrollo de software para sistemas empotrados. Herramientas de programación y técnicas de depuración.

2. El hardware de un sistema empotrado. Elementos básicos de un sistema empotrado. Tipos de procesadores empotrados. Memorias. Dispositivos de E/S. Buses digitales. Redes de comunicaciones cableadas o inalámbricas.

3. Desarrollo de software para empotrados. Introducción. Esquema host-target. Soportes de ejecución: núcleos y sistemas operativos empotrados de tiempo real. Generación de imágenes firmware (BSPs). Herramientas de programación para sistemas empotrados: Depuradores, emuladores, ensambladores, compiladores, o compiladores cruzados. Técnicas de depuración. Soporte que ofrecen los distintos lenguajes de programación para el desarrollo de aplicaciones empotradas.

4. Técnicas de programación a bajo nivel del empotrado. Manejo de E/S. Manejo de interrupciones y excepciones. Gestión de memoria. Relojes y temporizadores: Watchdog. Diseño de driver de dispositivo: modelo sincrónico y asíncrónico.

5. Programación de tiempo real. Tareas periódicas, aperiódicas y esporádicas. Modelos de programación. Planificación de sistemas de tiempo real: ejecutivos cíclico y basados en prioridad. Análisis temporal de tareas de tiempo real: análisis basado en el tiempo de respuesta.



Mecanismos de comunicación y sincronización entre tareas.

6. Redes de comunicaciones en entornos empotrados. Redes cableadas e inalámbricas. Protocolos de comunicaciones. Modelos de programación distribuidas (paso de mensajes, objetos remotos, agentes).

7. Java para el desarrollo de aplicaciones empotradas. ¿Por qué Java? Aspectos principales de Java. Restricciones para el desarrollo de aplicaciones empotradas. Especificaciones Java: JavaME, RTSJ, SCJ. Modelo de ejecución basado en la máquina virtual (JVM): motor de ejecución, carga dinámica, recolector de basura. Plataformas empotradas basadas en Java.

8. Desarrollo de proyectos software para entornos empotrados. Métodos y técnicas avanzadas para el desarrollo de proyectos empotrados. Particionamiento Hardware/Software. Arquitectura software de un sistema. Metodología del proceso de desarrollo del sistema empotrado basado en el lenguaje UML (COMET, OCTOPUS, UML-RT, ...). Etapas del proceso de desarrollo (Análisis de los requerimientos, diseño e integración con el hardware, implementación, pruebas y mantenimiento del sistema). Ciclos de vida para el desarrollo de empotrados (desarrollo rápido de aplicaciones, prototipado rápido, ...).

9. Aplicaciones actuales y nuevas vías de investigación. Redes de sensores, automatización industrial, inteligencia ambiental, instrumentación virtual, control distribuido, sensores inteligentes, mecatrónica, ...

1. Introduction to embedded systems. Specific requirements of embedded systems design: criticality, real-time constraints, cost per unit, energy consumption, ... Basic hardware of an embedded system. Comparison between different embedded architectures. Basic software of an embedded system. Software development for embedded systems. Programming tools and debugging techniques.

2. The hardware of an embedded system. Basic elements of an embedded system. Types of embedded processors. Memories. I/O Devices. Digital buses. Wired communications networks or wireless.

3. Development of software for embedded systems. Host-target scheme. Execution environment support: kernels and real time embedded operating systems. Generation of firmware images (BSPs). Programming tools for embedded systems: Debuggers, Emulators, assemblers, compilers, cross or compilers. Debugging techniques. Programming languages support for developing embedded applications.

4. Low level programming techniques for embedded systems. I/O device programming. Interrupt handling and exceptions. Memory management. Clocks and timers: Watchdog. Design of device driver: synchronous and asynchronous model.

5. Real Time programming. Periodic, aperiodic and sporadic tasks. Programming models. Scheduling of real time systems: cyclic executive and priority-based. Temporal Analysis of real time tasks: Response-Time Analysis. Mechanisms for communication and synchronization between tasks.

6. Networks in embedded environments. Wired and wireless networks. Communications protocols. Distributed programming models (message passing, remote objects, agents).

7. Java for developing embedded applications. Why Java? Main aspects of Java. Restrictions for



developing embedded applications. Java Specifications: JavaME, RTSJ, SCJ. Execution model based on virtual machine (JVM): runtime, dynamic loading, garbage collector. Embedded platforms based on Java.

8. Software development projects for embedded environments. Methods and advanced techniques for the development of embedded projects. Hardware/Software partitioning. Software architecture of a system. Embedded system development methodology based on UML (COMET, OCTOPUS, UML-RT, ...). Stages of the development process (requirements analysis, design and integration with the hardware, implementation, testing and maintenance of the system). Life cycles for the development of embedded system (rapid application development, rapid prototyping, ...).

9. Current applications and new research lines. Sensor networks, industrial automation, ambient intelligence, virtual instrumentation, distributed control, smart sensors, mechatronics,

PRÁCTICO

Seminarios/Talleres

- Metodología para el desarrollo de un proyecto para sistema empotrado.
- Guía de desarrollo de aplicaciones en Arduino.
- Desarrollo de aplicaciones en POSIX.

Prácticas de Laboratorio:

Práctica 1. Desarrollo de aplicaciones sobre microcontroladores con recursos limitados: Arduino.

Práctica 2. Desarrollo de Aplicaciones.POSIX en entornos de ejecución de tiempo real sobre microprocesadores empotrados: Programación de un robot..

Práctica 3. Programación de sistemas empotrados con Java.

BIBLIOGRAFÍA

BIBLIOGRAFÍA FUNDAMENTAL

- Software Engineering for Embedded Systems. Robert Oshana y Mark Kraeling. 2nd Edition. Newnes, 2019

- Embedded Systems Design. Steve Heath. 2nd Edition. Newnes, 2003.

- Real-Time Concepts for Embedded Systems. Qing Li and Caroline Yao. CMP Books, 2003.

- Software Engineering for Real-Time Systems. Jim Cooling. Addison Wesley, 2002.

- Computers As Components: Principles of Embedded Computing Systems Design. Wayne Wolf. Morgan Kaufman Publisher, 2001.

- Designing Concurrent, Distributed, and Real-Time Applications with UML. H. Gommaa. Prentice Hall, 2000.



- Real Time UML. B. Powell Douglass. Third Edition. Prentice Hall, 2004

BIBLIOGRAFÍA COMPLEMENTARIA

- Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers. Tammy Noergaard. Newnes, 2005.

- Embedded Systems Design: An Introduction to Processes, Tools and Techniques. Arnold Berger CMP Books, 2001

ENLACES RECOMENDADOS

Como apoyo a la docencia se usará la Plataforma de Recursos de Apoyo a la Docencia PRADO2 de la Universidad de Granada: <https://prado.ugr.es/>

METODOLOGÍA DOCENTE

- MD01 Lección magistral/expositiva
- MD02 Sesiones de discusión y debate
- MD03 Resolución de problemas y estudio de casos prácticos
- MD04 Prácticas de laboratorio o clínicas
- MD05 Seminarios
- MD06 Ejercicios de simulación
- MD07 Análisis de fuentes y documentos
- MD08 Realización de trabajos en grupo
- MD09 Realización de trabajos individuales

EVALUACIÓN (instrumentos de evaluación, criterios de evaluación y porcentaje sobre la calificación final)

EVALUACIÓN ORDINARIA

El artículo 18 de la Normativa de Evaluación y Calificación de los Estudiantes de la Universidad de Granada establece que la convocatoria ordinaria estará basada preferentemente en la evaluación continua del estudiante, excepto para quienes se les haya reconocido el derecho a la evaluación única final.

Se realizará una evaluación continua del trabajo del estudiante, valorando tanto los conocimientos adquiridos como las competencias alcanzadas.

Modalidad presencial:

Para la evaluación en modalidad presencial se tendrá en cuenta:

1. Asistencia y participación activa en el Aula durante la impartición del curso: interés del estudiante, respuesta a las preguntas planteadas por el profesor durante la sesión, etc.



2. Participación en foros de debate o de recogida de información. Se propondrán distintos temas de debate para que se puedan discutir en el aula o a través de la plataforma docente.

3. Resolución de ejercicios o entrega de trabajos, informes, a través de la plataforma docente. Se propondrán distintas actividades tanto teóricas como prácticas durante la impartición del curso cuya entrega se realizará a través de la plataforma docente PRADO.

4. Cuestionarios on-line. Se completarán diferentes cuestionarios durante la impartición del curso para asentar los conocimientos adquiridos.

Se tratará de realizar actividades adaptadas a la temática de los trabajos fin de máster que están realizando los estudiantes.

Se pedirá la entrega en tiempo y forma de las actividades propuestas a través de la plataforma PRADO.

La evaluación final se calculará considerando los siguientes porcentajes:

- Asistencia y participación activa en el aula: 10%
- Participación en foros de debate o de recogida de información: 10%
- Resolución de ejercicios o entrega de trabajos, informes, a través de la plataforma docente: 40%
- Cuestionarios on-line: 40%

Modalidad virtual:

Para la evaluación en modalidad virtual se tendrá en cuenta:

1. Participación activa a través de la plataforma: se realizarán distintas preguntas durante la impartición de los contenidos del curso.

2. Participación en foros de debate o de recogida de información. Se propondrán distintos temas de debate para que se puedan discutir en el aula o a través de la plataforma docente.

3. Resolución de ejercicios o entrega de trabajos, informes, a través de la plataforma docente. Se propondrán distintas actividades tanto teóricas como prácticas durante la impartición del curso cuya entrega se realizará a través de la plataforma docente PRADO.

4. Cuestionarios on-line. Se completarán diferentes cuestionarios durante la impartición del curso para asentar los conocimientos adquiridos.

Se tratará de realizar actividades adaptadas a la temática de los trabajos fin de máster que están realizando los estudiantes.

Se pedirá la entrega en tiempo y forma de las actividades propuestas a través de la plataforma PRADO.

La evaluación final se calculará considerando los siguientes porcentajes:

- Participación activa a través de la plataforma: 10%



- Participación en foros de debate o de recogida de información: 10%
- Resolución de ejercicios o entrega de trabajos, informes, a través de la plataforma docente: 40%
- Cuestionarios on-line: 40%

Aunque los criterios de evaluación en las dos modalidades son similares, las actividades a realizar para cada modalidad pueden diferir, al contar con herramientas distintas tanto para la adquisición de los conocimientos como para la evaluación.

A continuación se especifican las actividades formativas previstas así como su temporalización dependiendo de la modalidad de estudio:

Actividades formativas	Hora lectivas	Modalidad Semipresencial		Modalidad virtual
		Horas presenciales	Presencialidad	Horas virtuales
Clases teóricas	10	10	100%	10
Clases prácticas	8	8	100%	8
Trabajos Tutorizados	2	2	100%	2
Tutorías	8	8	100%	8
Evaluación	2	2	100%	2
Trabajo Autónomo	45	45		45
Total	75	75		75

Se llevarán a cabo sesiones orales para el control, evaluación y seguimiento de todos los alumnos. Por una parte, se realizará una defensa oral de algunas entregas así como contestar a algunos cuestionarios planteados.

Se realizará el seguimiento de las incidencias y dificultades que tengan los estudiantes en la modalidad presencial y virtual a través de la herramienta PRADO

EVALUACIÓN EXTRAORDINARIA

El artículo 19 de la Normativa de Evaluación y Calificación de los Estudiantes de la Universidad de Granada establece que los estudiantes que no hayan superado la asignatura en la convocatoria ordinaria dispondrán de una convocatoria extraordinaria. A ella podrán concurrir todos los estudiantes, con independencia de haber seguido o no un proceso de evaluación continua. De esta forma, el estudiante que no haya realizado la evaluación continua tendrá la posibilidad de obtener el 100% de la calificación mediante la realización de una prueba y/o trabajo.

EVALUACIÓN ÚNICA FINAL

El artículo 8 de la Normativa de Evaluación y Calificación de los Estudiantes de la Universidad de Granada establece que podrán acogerse a la evaluación única final, el estudiante que no pueda cumplir con el método de evaluación continua por causas justificadas.

Para acogerse a la evaluación única final, el estudiante, en las dos primeras semanas de impartición de la asignatura o en las dos semanas siguientes a su matriculación si ésta se ha producido con posterioridad al inicio de las clases o por causa sobrevenidas. Lo solicitará, a través del procedimiento electrónico, a la Coordinación del Máster, quien dará traslado al profesorado correspondiente, alegando y acreditando las razones que le asisten para no poder seguir el





sistema de evaluación continua.

La evaluación en tal caso consistirá en la realización de una prueba, un trabajo, y/o las actividades propuestas en la evaluación continua.

