Proyecto/Guía docente de la asignatura

Asignatura	SISTEMAS DE TIEMPO REAL		
Materia	DOMINIOS ESPECÍFICOS DE APLICACIÓN		
Módulo	MATERIAS ESPECÍFICAS DE LA MENCIÓN EN TELEMÁTICA		
Titulación	GRADO EN INGENIERÍA DE TECNOLOGÍAS ESPECÍFICAS DE TELECOMUNICACIÓN		
Plan	512	**Código**	46673
Periodo de impartición	2º CUATRIMESTRE	**Tipo/Carácter**	OPTATIVA DE LA MENCIÓN
Nivel/Ciclo	GRADO	**Curso**	4º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	MARIO MARTÍNEZ ZARZUELA DAVID GONZÁLEZ ORTEGA		
Datos de contacto (E-mail, teléfono…)	TELÉFONO: 98342300 ext. 5702, ext. 5552 E-MAIL: mario.martinez@uva.es, david.gonzalez.ortega@uva.es DESPACHOS: 2D006,2D007		
Departamento	TEORÍA DE LA SEÑAL Y COMUNICACIONES E INGENIERÍA TELEMÁTICA		
Fecha de revisión por el Comité de Título	26 de junio de 2023		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

Los *Sistemas de Tiempo Real* son aquellos sistemas en los que las restricciones del tiempo de respuesta son tan importantes como la corrección de la respuesta en sí misma. Generalmente, estos sistemas deben interactuar con los elementos físicos de su entorno y tienen aplicación en áreas como los sistemas empotrados, en sistemas de comunicación, en robótica y el control de procesos, entre otras. La cantidad de información que reciben y deben procesar estos sistemas se ha incrementado de manera muy relevante para ciertas aplicaciones. En este caso, resulta fundamental conocer las bases del funcionamiento de los microprocesadores y la optimización de los algoritmos de decisión mediante técnicas de programación paralela.

En esta asignatura se enmarca el campo de los Sistemas de Tiempo Real dentro de las expectativas del alumno que cursa el plan de estudios del Grado en Ingeniería de Tecnologías Específicas de Telecomunicación, en la Mención en Ingeniería Telemática. Por este motivo, se estudian los aspectos más generales de la disciplina, para más tarde abordar con detenimiento herramientas y entornos de desarrollo de especial relevancia para su perfil. En esta asignatura se realizan prácticas de programación con sistemas embebidos con sistema operativo (Raspberry Pi, Jetson Nano), que permiten el despliegue rápido de aplicaciones para Internet de las Cosas (IoT) y se aprenden las bases del procesamiento en paralelo con Unidades Gráficas de Procesamiento (GPUs), que se emplean cada vez más en sistemas de tiempo real, tales como en sistemas de ayuda a la conducción o sistemas de conducción autónoma. En la asignatura el alumno utiliza el lenguaje C, que ya ha estudiado a lo largo de su carrera y se introduce en la programación con Python y, en menor medida, C para CUDA.

1.2 Relación con otras materias

Está asignatura está relacionada con las materias y asignaturas siguientes del Grado en Ingeniería de Tecnologías Específicas de Telecomunicación:

- Materia Informática, perteneciente a las Materias Instrumentales. Asignatura relacionada: *Programación*.
- Materia Fundamentos de Ordenadores, perteneciente a las Materias Básicas de Telecomunicaciones. Asignatura relacionada: *Fundamentos de Ordenadores y Sistemas Operativos*.
- Materia Electrónica Digital. Asignaturas relacionadas: Circuitos Electrónicos Digitales, Sistemas Electrónicos Basados en Microprocesador.
- Materia Fundamentos de Sistemas Software perteneciente a las Materias Básicas de Telecomunicaciones. Asignatura relacionada: *Ingeniería de Sistemas Software*.

1.3 Prerrequisitos

- Conocimientos básicos de Programación en C.
- Conocimientos básicos de Arquitectura de Ordenadores.
- Conocimientos básicos de Sistemas Operativos.
- Conocimientos básicos de Electrónica Digital.
- Conocimientos básicos de Ingeniería de Sistemas Software.

2. Competencias

2.1 Generales

- **GBE2**. Capacidad para aplicar métodos analíticos y numéricos para el análisis de problemas en el ámbito de la ingeniería técnica de Telecomunicación.
- **GBE3**. Capacidad para resolver problemas con iniciativa, creatividad y razonamiento crítico.
- **GBE4**. Capacidad para diseñar y llevar a cabo experimentos, así como analizar e interpretar datos.
- **GBE5**. Capacidad para elaborar informes basados en el análisis crítico de la bibliografía técnica y de la realidad en el campo de su especialidad.
- **GE2.** Capacidad para trabajar en un grupo multidisciplinar y multilingüe, responsabilizándose de la dirección de actividades objeto de los proyectos del ámbito de su especialidad y consiguiendo resultados eficaces.
- **GE3**. Capacidad para desarrollar metodologías y destrezas de aprendizaje autónomo eficiente para la adaptación y actualización de nuevos conocimientos y avances científicos.
- **GC1**. Capacidad de organización, planificación y gestión del tiempo.
- **GC2**. Capacidad para comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.
- **GC3**. Capacidad para trabajar en cualquier contexto, individual o en grupo, de aprendizaje o profesional, local o internacional, desde el respeto a los derechos fundamentales, de igualdad de sexo, raza o religión y los principios de accesibilidad universal, así como la cultura de paz.

2.2 Específicas

- **B14.** Conocer los conceptos básicos de la arquitectura de ordenadores y de los servidores, así como los principios de los sistemas operativos.
- **T1.** Capacidad para aprender de manera autónoma nuevos conocimientos y técnicas adecuados para la concepción, el desarrollo o la explotación de sistemas y servicios de telecomunicación.
- **T2.** Capacidad de utilizar aplicaciones informáticas (ofimáticas, bases de datos, cálculo avanzado, gestión de proyectos, visualización, etc.) para apoyar el desarrollo y explotación de redes, servicios y aplicaciones de telecomunicación y electrónica.
- **T3.** Capacidad para utilizar herramientas informáticas de búsqueda de recursos bibliográficos o de información relacionada con las telecomunicaciones y la electrónica.
- **T12.** Conocimiento y utilización de los conceptos de arquitectura de red, protocolos e interfaces de comunicaciones.
- **TEL7.** Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas.

3. Objetivos

Al finalizar la asignatura el alumno deberá ser capaz de:

- Comprender la naturaleza específica de los sistemas de tiempo real.
- Conocer los requisitos de los lenguajes de programación para el diseño de sistemas de tiempo real.
- Adaptar los principales mecanismos que poseen los lenguajes de programación para la implementación de servicios y aplicaciones de tiempo real.
- Identificar, implementar y valorar la importancia de la concurrencia de procesos en sistemas de control software y afrontar la problemática derivada.
- Conocer los modelos básicos de programación con procesos e hilos y su aplicación a sistemas concretos de tiempo real.
- Desarrollar sistemas de tiempo real no estrictos usando ordenadores embebidos tipo PC.
- Entender las diferencias entre paralelismo a nivel de tarea y paralelismo a nivel de datos.
- Conocer las bases de la programación de procesadores masivamente paralelos.

4. Contenidos y/o bloques temáticos

Bloque 1: "Introducción general"

Carga de trabajo en créditos ECTS: 1.2

a. Contextualización y justificación

Este bloque temático introduce y relaciona los contenidos que se estudiarán en los siguientes bloques temáticos, a fin de que el alumno adquiera una visión global de la asignatura ya en las primeras semanas. Con esta intención, se tratarán superficialmente los conceptos y aspectos tecnológicos de toda la asignatura, facilitando al alumno la tarea de asociar entre sí los distintos contenidos, como paso previo al estudio detallado de cada uno de ellos. Así mismo, se realiza un repaso de aspectos avanzados de programación en lenguaje C, que servirá para adentrarse en el siguiente bloque en la programación con llamadas al sistema tipo POSIX en Unix/Linux. Adicionalmente, se realizará una introducción al lenguaje de programación interpretado Python.

b. Objetivos de aprendizaje

Al finalizar este bloque temático, el alumno deberá ser capaz de:

- Comprender la naturaleza específica de los sistemas de tiempo real.
- Conocer los requisitos de los lenguajes de programación y los sistemas operativos para el diseño de sistemas de tiempo real.
- Realizar programas con características avanzadas del lenguaje C.
- Realizar programas básicos en lenguaje Python.

c. Contenidos

Este bloque incluye los temas y sus secciones correspondientes que se indican a continuación:

TEMA 0: Presentación de la asignatura

0.1 Descripción de competencias y contenidos.
0.2 Objetivos de aprendizaje.
0.3 Herramientas disponibles.
0.4 Sistema de evaluación.

TEMA 1: Introducción a los sistemas de tiempo real (STR)

1.1 Conceptos y aspectos generales.
1.2 Sistemas operativos y lenguajes de programación para STR.
1.3 Diseño de sistemas de tiempo real.
1.4 Ejemplos de STR.

TEMA 2: Programación avanzada en lenguaje C

2.1 Programación incremental utilizando make.
2.2 Punteros multidimensionales y reserva dinámica de memoria.
2.3 Punteros a funciones.
2.4 Ejercicios de programación.

TEMA 3: Introducción al lenguaje Python

3.1 Introducción al lenguaje y tipos de datos.
3.2 Funciones y programación orientada a objetos.
3.3 Manejo de E/S y excepciones.

3.4 Módulos y paquetes en Python.

3.5 Ejercicios de programación.

d. Métodos docentes

- Clases prácticas en aula.
- Prácticas en laboratorio.

e. Plan de trabajo

Véase Anexo I. Plan de Trabajo.

f. Evaluación

La evaluación de la adquisición de competencias se basará en:

- Capacidad para identificar los componentes fundamentales de un sistema de tiempo real.
- Capacidad para desarrollar programas en C con funciones POSIX.
- Capacidad para desarrollar programas básicos en Python.
- Valoración de la actitud y participación del alumno en las actividades formativas.

g Material docente

g.1 Bibliografía básica

- Burns, A., Wellings, A., Real-Time Systems and Programming Languages (4th Edition), Addison Wesley, 2009. ISBN: 978-0321417459. https://buc-uva.alma.exlibrisgroup.com/leganto/public/34BUC_UVA/citation/5065538700005774?auth=SAML
- Jiménez García, L.M., Puerto Manchón, R., Reinoso García, O., Sistemas informáticos de tiempo real. Universidad Miguel Hernández, 2000. ISBN: 84-95315-51-3.
- Kernighan, B.W., Pike, R., El entorno de programación Unix, Prentice Hall, 1995. ISBN: 978-9688800676.
- Marzal, A. y Gracia, I., Introducción a la programación con Python, Universitat Jaume I. ISBN: 978-84-692-5869-9. Disponible en: http://repositori.uji.es/xmlui/bitstream/handle/10234/102653/s93.pdf

g.2 Bibliografía complementaria

- Welling, A., Concurrent and Real-Time Programming in Java, John Wiley & Sons, 2004. ISBN: 978-0470844373.
- Buhr, R.J.A., An Introduction to Real-Time Systems, Prentice Hall, 1998. ISBN: 978-0136060703.
- Tanenbaum, A. S., Sistemas operativos modernos, Prentice-Hall, 2009. ISBN: 978-6074420463.
- Stalling, W., Sistemas Operativos: Principios de diseño e interioridades (5ª Edición), Prentice Hall, 2006. ISBN: 978-8420544625. https://buc-uva.alma.exlibrisgroup.com/leganto/public/34BUC_UVA/citation/5065589240005774?auth=SAML

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), …)

En caso de que la formación tuviese que ser desarrollada en modalidad online por mandato de las autoridades competentes, se utilizarán vídeos con la presentación y explicación de los contenidos.

h. Recursos necesarios

Serán necesarios los siguientes recursos, todos ellos facilitados por la Universidad o el profesor:

- Entorno de trabajo en la plataforma Moodle.
- Documentación de apoyo.
- Laboratorio con sistema operativo Linux.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1	3 semanas. Semanas 1 a 3.

Bloque 2: "Concurrencia, comunicación y sincronización en sistemas Unix/Linux"

Carga de trabajo en créditos ECTS: 1.8

a. Contextualización y justificación

En este bloque temático se estudian los aspectos teóricos necesarios sobre sistemas concurrentes y los mecanismos asociados. Se estudiará la gestión y planificación de procesos e hilos del sistema operativo y los métodos disponibles para crear, comunicar y sincronizar procesos e hilos mediante el desarrollo de aplicaciones. Con tal fin, se introducen en este tema las llamadas al sistema con el estándar POSIX de programación en C y los módulos de Python para conseguir concurrencia.

b. Objetivos de aprendizaje

Al finalizar este bloque temático, el alumno deberá ser capaz de:

- Adaptar los principales mecanismos que poseen los lenguajes de programación para la implementación de servicios y aplicaciones de tiempo real.
- Identificar, implementar y valorar la importancia de la concurrencia de procesos en sistemas de control software y afrontar la problemática derivada.
- Conocer los modelos básicos de programación con procesos e hilos y su aplicación a sistemas concretos de tiempo real.
- Entender la necesidad de ciertas estrategias para la mejora del rendimiento de programas.

c. Contenidos

Este bloque incluye los temas y sus secciones correspondientes que se indican a continuación:

TEMA 4: Programación de procesos e hilos en C (POSIX)

4.1 Programación de procesos en C (POSIX).

4.2 Programación de hilos en C (POSIX).

TEMA 5: Programación de procesos e hilos en Python

5.1 Programación de procesos e hilos en Python.

d. Métodos docentes

- Clases prácticas en aula.

- Prácticas en laboratorio.
- Seminarios.

e. Plan de trabajo

Véase Anexo I. Plan de Trabajo.

f. Evaluación

La evaluación de las competencias adquiridas se realiza mediante:

- Pruebas de laboratorio en las que se evalúan la capacidad para desarrollar programas en C con funciones POSIX y la capacidad para desarrollar programas básicos en Python usando módulos que permitan concurrencia.
- Valoración de la actitud y participación del alumno en las actividades formativas.

g Material docente

g.1 Bibliografía básica

- Jiménez García, L.M., Puerto Manchón, R., Reinoso García, O., Sistemas informáticos de tiempo real. Universidad Miguel Hernández, 2000. ISBN: 84-95315-51-3. https://buc-uva.alma.exlibrisgroup.com/leganto/public/34BUC_UVA/citation/5139615480005774?auth=SAML

g.2 Bibliografía complementaria

- Mitchell, M., Oldham, J. and Samuel, A, Advanced Linux Programming, New Riders Publishing, 2001. ISBN 0-7357-1043-0. Disponible en: http://advancedlinuxprogramming.com. https://buc-uva.alma.exlibrisgroup.com/leganto/public/34BUC_UVA/citation/5065595420005774?auth=SAML
- Kernighan, B.W., Pike, R., El entorno de programación Unix, Prentice Hall, 1995. ISBN: 978-9688800676.
- Stalling, W. Sistemas Operativos: Principios de diseño e interioridades (5ª Edición), Prentice Hall, 2006. ISBN: 978-8420544625. https://buc-uva.alma.exlibrisgroup.com/leganto/public/34BUC_UVA/citation/5139628870005774?auth=SAML
- Welling, A., Concurrent and Real-Time Programming in Java, John Wiley & Sons, 2004. ISBN: 978-0470844373.
- Palach, J. Parallel Programming with Python, Packt Publishing Ltd, 2014. ISBN: 978-1-78328-839-7.
- Dimitriadis Y. y Díaz Pernas F. J. Introducción práctica a la administración de sistemas en Internet. Servicio de Publicaciones de la Universidad de Valladolid, 1998. https://buc-uva.alma.exlibrisgroup.com/leganto/public/34BUC_UVA/citation/5139630750005774?auth=SAML

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), …)

En caso de que la formación tuviese que ser desarrollada en modalidad online por mandato de las autoridades competentes, se utilizarán vídeos con la presentación y explicación de los contenidos.

h. Recursos necesarios

Serán necesarios los siguientes recursos, todos ellos facilitados por la Universidad o el profesor:

- Entorno de trabajo en la plataforma Moodle.
- Documentación de apoyo.
- Laboratorio con sistemas operativos Windows y Linux.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 2	4 semanas y media. Semanas 4 a 7 y media.

Bloque 3: "Programación de sistemas embebidos de placa única"

Carga de trabajo en créditos ECTS: 3.0

a. Contextualización y justificación

En este bloque temático se realiza una introducción al funcionamiento del sistema embebido tipo PC Raspberry Pi y el sistema Jetson Nano. Se estudian los aspectos teóricos sobre procesadores masivamente paralelos tipo GPU (Graphics Processing Unit) y su utilización en Inteligencia Artificial. Se proporcionará una visión histórica de la evolución de los procesadores gráficos hacia procesadores de cómputo general y de los principales campos de investigación y desarrollo en los que se están utilizando. Se introducirá al alumno en la programación de estos sistemas mediante los lenguajes de programación C y Python (CUDA).

b. Objetivos de aprendizaje

Al finalizar este bloque temático, el alumno deberá ser capaz de:

- Instalar el sistema operativo en un ordenador de bajo coste Raspberry Pi / Jetson Nano
- Aprender las bases de la programación de redes neuronales con Inteligencia Artificial en sistemas embebidos
- Entender las diferencias entre paralelismo a nivel de tarea y paralelismo a nivel de datos.
- Conocer las bases de la programación de procesadores masivamente paralelos.

c. Contenidos

Este bloque incluye los temas y sus secciones correspondientes que se indican a continuación:

TEMA 6: Introducción a Raspberry Pi y Jetson Nano

6.1 Instalación y preparación de Raspberry Pi y Jetson Nano

6.2 Introducción al GPIO de Raspberry Pi y Jetson Nano

TEMA 7: Introducción a la programación de redes neuronales en sistemas embebidos

7.1 Introducción al aprendizaje automático sobre dispositivos embebidos

7.2 Entrenamiento y despliegue de modelos neuronales

TEMA 8: Sensores y actuadores para Raspberry Pi y Jetson Nano

8.1 Conceptos y aspectos generales.

8.2 Ejemplos de programación con sensores y actuadores.

TEMA 9: Introducción al Procesamiento Paralelo en GPU

9.1 Historia y evolución.

9.2 Desarrollo de aplicaciones.

d. Métodos docentes

- Clases prácticas en aula.
- Prácticas en laboratorio.
- Prácticas en nube de computación.

e. Plan de trabajo

Véase Anexo I. Plan de Trabajo.

f. Evaluación

La evaluación de la adquisición de competencias se basará en:

- Capacidad para identificar los componentes internos de una GPU.
- Capacidad para desarrollar programas en CUDA para C y Python con procesamiento en GPU.

g Material docente

g.1 Bibliografía básica

- Sanders, J. and Kandrot, E., CUDA by Example: An Introduction to General-Purpose GPU Programming, Addison-Wesley Professional, 2010. ISBN: 978-0131387683.
- Kirk, D.B., Mei, W., Programming Massively Parallel Processors, Second Edition: A Hands-on Approach, Morgan Kaufmann, 2012. ISBN: 978-0124159921. https://buc-uva.alma.exlibrisgroup.com/leganto/public/34BUC_UVA/citation/5065601200005774?auth=SAML
- Cox, T., Raspberry Pi Cookbook for Python Programmers, Packt Publishing, 2014. ISBN 978-1-84969-662-3.

g.2 Bibliografía complementaria

g.3 Otros recursos telemáticos (píldoras de conocimiento, blogs, videos, revistas digitales, cursos masivos (MOOC), …)

En caso de que la formación tuviese que ser desarrollada en modalidad online por mandato de las autoridades competentes, se utilizarán vídeos con la presentación y explicación de los contenidos.

h. Recursos necesarios

Serán necesarios los siguientes recursos, todos ellos facilitados por la Universidad o el profesor:

- Entorno de trabajo en la plataforma Moodle.
- Documentación de apoyo.
- Laboratorio con sistemas operativos Windows y Linux.

- Ordenador Raspberry Pi, Jetson Nano y placa de aprendizaje GTI-clip.
- Nube de computación con GPUs dedicadas.

i. Temporalización

CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 3	7 semanas y media. Semanas 7 y media a 15.

5. Métodos docentes y principios metodológicos

El estudiante adquirirá los conocimientos teórico prácticos en clases magistrales participativas, y mediante la lectura y el visionado de materiales preparados o enlazados por los profesores. Los conocimientos de carácter eminentemente práctico se adquieren a través de prácticas guiadas de programación con autocorrección, que permiten a los estudiantes adquirir nuevos conocimientos a partir de las bases teóricas.

Durante el tiempo que duren estas actividades, se ofrecerán al estudiante tiempos de tutoría síncrona o asíncrona. En el primer caso se emplearán entornos presenciales o virtuales, tales como videoconferencia y chat. En el segundo caso se emplearán foros de discusión abiertos en el Campus Virtual para tal efecto o sistemas de mensajería electrónica.

Se priorizará la evaluación continua de competencias, empleando los siguientes instrumentos: pruebas de programación que serán realizadas individualmente por los estudiantes y evaluadas por el profesor de forma asíncrona, y trabajos en grupo con el enfoque metodológico del Aprendizaje Basado en Proyectos y revisión entre pares.

6. Tabla de dedicación del estudiante a la asignatura

ACTIVIDADES PRESENCIALES o PRESENCIALES A DISTANCIA[1]	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases prácticas de aula (A)	15	Estudio y trabajo autónomo individual	90
Laboratorios (L)	30	Estudio y trabajo autónomo grupal	0
Seminarios (S)	10		
Evaluación (fuera del periodo oficial de exámenes)	5		
Total presencial	**60**	Total no presencial	**90**
		TOTAL presencial + no presencial	**150**

[1] Actividad presencial a distancia en este contexto es aquella en que el grupo sigue por videoconferencia la clase impartida por el profesor en el horario publicado para la asignatura.

7. Sistema y características de la evaluación

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
Pruebas objetivas sobre funcionamiento de los desarrollos realizados en los ejercicios de laboratorio de los bloques 1, 2 y 3.	90% *(15%, 30% 25%, 20%)*	Con la entrega de cada enunciado se informará al alumno de cada uno de los aspectos susceptibles de ser evaluados y los criterios a emplear.
Actitud y participación.	10%	Valoración de la actitud en el laboratorio y participación activa en las clases y el campus virtual.

CRITERIOS DE CALIFICACIÓN

• **Convocatoria ordinaria:**
  ○ La nota final de la convocatoria ordinaria se calculará de acuerdo a los dos instrumentos y sus pesos reflejados en la tabla anterior.
• **Convocatoria extraordinaria:**
  ○ El alumno podrá elegir guardar la nota obtenida en la convocatoria ordinaria en los instrumentos de evaluación que desee, salvo la de actitud y participación, y siempre y cuando haya obtenido en ellas una nota relativa de al menos 4 puntos sobre 10.
  ○ Para la evaluación del resto de competencias tendrá que realizar un examen.

8. Consideraciones finales

El Anexo I con el Plan de Trabajo estará disponible con anterioridad al comienzo de la asignatura.