



Universidad
Politécnica
de Cartagena



Guía docente de la asignatura

Programación de Sistemas en Tiempo Real

Titulación: Grado en Ingeniería Electrónica Industrial y Automática

CSV:	bvlXDcmZNDohlYTso5JTtlxo9	Fecha:	16/01/2019 13:20:31	
Normativa:	Este documento es copia auténtica imprimible de un documento administrativo firmado electrónicamente y archivado por la Universidad Politécnica de Cartagena.			
Firmado Por:	Universidad Politécnica de Cartagena - Q8050013E			
Url Validación:	https://validador.upct.es/csv/bvlXDcmZNDohlYTso5JTtlxo9	Página:	1/13	

1. Datos de la asignatura

Nombre	Programación de Sistemas en Tiempo Real				
Materia	Informática Industrial				
Módulo	Materias específicas				
Código	507103008				
Titulación	Grado en Ingeniería Electrónica Industrial y Automática				
Plan de estudios	2010				
Centro	Escuela Técnica Superior de Ingeniería Industrial				
Tipo	Obligatoria				
Periodo lectivo	Cuatrimstral	Cuatrimestre	C2	Curso	3
Idioma	Español				
ECTS	4,5	Horas / ECTS	30	Carga total de trabajo (horas)	135

* Todos los términos marcados con un asterisco que aparecen en este documento están definidos en *Referencias para la actividad docente en la UPCT y Glosario de términos:*

<http://repositorio.bib.upct.es/dspace/bitstream/10317/3330/1/isbn8469531360.pdf>

2. Datos del profesorado

Profesor responsable	Diego Alonso Cáceres		
Departamento	Tecnologías de la Información y las Comunicaciones		
Área de conocimiento	Lenguajes y Sistemas Informáticos		
Ubicación del despacho	2ª Planta Antiguo Cuartel de Antiguones, despacho 25		
Teléfono	968 325341	Fax	968 325973
Correo electrónico	diego.alonso@upct.es		
URL / WEB	Aula Virtual UPCT		
Horario de atención / Tutorías	Según establezca el profesor		
Ubicación durante las tutorías	Despacho del profesor		

Titulación	Ingeniero Industrial (Universidad Politécnica Valencia) Doctorado con Mención Europea (UPCT)
Vinculación con la UPCT	Profesor Titular de Universidad
Año de ingreso en la UPCT	2004
Nº de quinquenios (si procede)	2
Líneas de investigación (si procede)	Ingeniería del Software para Sistemas Reactivos, Robótica, Desarrollo de Software Dirigido por Modelos, Desarrollo Basado en Componentes, Sistemas de Tiempo-Real, Programación de Agentes
Nº de sexenios (si procede)	2
Experiencia profesional (si procede)	
Otros temas de interés	

Profesor (tardes)	Jose Miguel Morales Illán		
Departamento	Tecnologías de la Información y las Comunicaciones		
Área de conocimiento	Lenguajes y Sistemas Informáticos		
Ubicación del despacho	2ª Planta Antiguo Cuartel de Antigones, despacho 28		
Teléfono	968 326460	Fax	968 325973
Correo electrónico	josemiguel.morales@upct.es		
URL / WEB	Aula Virtual UPCT		
Horario de atención / Tutorías	Lunes de 18:00 a 20:00 y martes de 16:00 a 18:00		
Ubicación durante las tutorías	Despacho número 28, edificio Antigones, zona departamental, Campus Muralla del Mar.		

Titulación	Dr. Informática
Vinculación con la UPCT	Profesor Asociado
Año de ingreso en la UPCT	2008
Nº de quinquenios (si procede)	--
Líneas de investigación (si procede)	Ingeniería del Software, Sistemas Reactivos, Robótica
Nº de sexenios (si procede)	--
Experiencia profesional (si procede)	18 años en empresas como programador y analista de sistemas
Otros temas de interés	

3. Descripción de la asignatura

3.1. Descripción general de la asignatura

La asignatura *Programación de Sistemas en Tiempo Real* se centra en la explicación de las características fundamentales de los sistemas de tiempo-real, su diseño y análisis. Un sistema de tiempo-real es aquel sistema informático en el que no sólo es necesario realizar correctamente el cálculo, sino que también tiene que ser realizado en el tiempo estipulado. Los sistemas de tiempo-real se utilizan en todo sistema de control que interactúa con el mundo, como microondas, sistemas ABS, industriales, etc. El núcleo de la asignatura lo forma la utilización de C++ para programar aplicaciones concurrentes, así como las características avanzadas de programación de C++. La asignatura tiene un carácter eminentemente práctico.

3.2. Aportación de la asignatura al ejercicio profesional

La formación del Ingeniero Industrial integra las habilidades propias de la Ingeniería con los métodos de las Matemáticas y la Informática para formular y construir modelos para el diseño, análisis, evaluación y predicción de sistemas. Particularmente, el Ingeniero en Electrónica Industrial y Automática debe ser capaz además de diseñar sistemas informáticos de control y conocer los problemas asociados a dichos sistemas. Estos son los objetivos fundamentales de la asignatura *Programación de Sistemas en Tiempo Real*. La formación aportada al alumno le capacita para poder extrapolar los conocimientos adquiridos a otras infraestructuras (lenguajes de programación, sistemas operativos, plataformas de desarrollo, etc.), de manera que las competencias adquiridas sean útiles en su desarrollo profesional a medio y largo plazo.

3.3. Relación con otras asignaturas del plan de estudios

Es necesario haber superado la asignatura *Informática Aplicada*, impartida en primer curso, para entender *Programación de Sistemas en Tiempo Real* y poder realizar las prácticas. Guarda una estrecha relación con la asignatura *Sistemas Basados en Microprocesador*, así como con el resto de asignaturas de programación y arquitectura de computadores que se imparten en 3º del Grado.

3.4. Incompatibilidades de la asignatura definidas en el plan de estudios

Ninguna.

3.5. Recomendaciones para cursar la asignatura

Es una asignatura eminentemente práctica, en la que se profundiza en los conceptos de programación en C introducidos en la asignatura *Informática Aplicada* de primero y se describe C++. Se recomienda encarecidamente al alumno que repase los conceptos básicos de la programación imperativa y estructurada de dicha asignatura.

3.6. Medidas especiales previstas

Se adoptarán las medidas especiales que especifique la E.T.S.I.I. en virtud de la normativa vigente y del estudio de cada caso particular. En cualquier caso, el alumno afectado debe contactar con el profesor al principio del curso.

4. Competencias y resultados del aprendizaje

4.1. Competencias básicas* del plan de estudios asociadas a la asignatura

CB2 - Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

4.2. Competencias generales del plan de estudios asociadas a la asignatura

G3 - Conocimiento en materias básicas y tecnológicas, que les capacite para el aprendizaje de nuevos métodos y teorías, y les dote de versatilidad para adaptarse a nuevas situaciones.

G4 - Capacidad de resolver problemas con iniciativa, toma de decisiones, creatividad, razonamiento crítico y de comunicar y transmitir conocimientos, habilidades y destrezas en el campo de la Ingeniería Industrial.

4.3. Competencias específicas* del plan de estudios asociadas a la asignatura

E28 - Conocimiento aplicado de informática industrial y comunicaciones

4.4. Competencias transversales del plan de estudios asociadas a la asignatura

T5 - Aplicar a la práctica los conocimientos adquiridos.

4.5. Resultados** del aprendizaje de la asignatura

Al finalizar con éxito esta asignatura, los estudiantes deben ser capaces de:

- (1) Conocer y aplicar los nuevos conceptos de C++11/C++14 y la librería estándar: colecciones, *strings*, entrada/salida, matemáticas, etc.
- (2) Conocer y aplicar los conceptos de orientación a objetos y excepciones en C++ para resolver problemas de tamaño medio.
- (3) Describir las principales características y problemas de la programación concurrente y el funcionamiento del planificador de un sistema operativo.
- (4) Conocer los algoritmos más utilizados para resolver los problemas asociados a los programas concurrentes y comunicación mediante memoria compartida.
- (5) Diseñar aplicaciones concurrentes utilizando hilos, mutex y variables de condición.

- (6) Describir las características de un sistema de tiempo-real, algoritmos de planificación y principales problemas que pueden aparecer en su diseño.
- (7) Resolver problemas de planificación y realización de cronogramas.

**** Véase también la *Guía de apoyo para la redacción, puesta en práctica y evaluación de los resultados del aprendizaje*, de ANECA:**

http://www.aneca.es/content/download/12765/158329/file/learningoutcomes_v02.pdf

5. Contenidos

5.1. Contenidos del plan de estudios asociados a la asignatura

Programación orientada a objetos. Programación concurrente. Características e infraestructura de los sistemas de tiempo real. Sistemas operativos y lenguajes de programación de tiempo real.

5.2. Programa de teoría (unidades didácticas y temas)

UD 1. Introducción a los sistemas de tiempo-real

Tema 0. Introducción a la asignatura

Tema 1. Introducción a los sistemas de tiempo-real

UD 2. El lenguaje de programación C++

Tema 2. Introducción a C++. C++ no es C (aunque se le parezca)

Tema 3. Librería estándar de C++: matemática, *strings*, contenedores e iteradores, algoritmos, entrada/salida

Tema 4. Programación orientada a objetos

UD 3. Programación concurrente

Tema 5. Introducción a la programación concurrente. Procesos e hilos. El planificador del sistema operativo. POSIX

Tema 6. Comunicación mediante variable compartida

UD 4. Planificación de sistemas de tiempo-real

Tema 7. Planificación

5.3. Programa de prácticas (nombre y descripción de cada práctica)

Sesiones de Aula de Informática:

Se desarrollarán 2 prácticas de aula de informática, donde los alumnos resolverán y probarán con la asistencia del profesor pequeños ejemplos guiados. Estos ejemplos tienen la complejidad suficiente para poder ser seguidos sin dificultad al tiempo que refuercen los conocimientos adquiridos en el desarrollo de las clases de teoría.


P1: programación orientada en objetos en C++ (4 sesiones)

P2: programación concurrente (3 sesiones)

Prevención de riesgos

La Universidad Politécnica de Cartagena considera como uno de sus principios básicos y objetivos fundamentales la promoción de la mejora continua de las condiciones de trabajo y estudio de toda la Comunidad Universitaria.

Este compromiso con la prevención y las responsabilidades que se derivan atañe a todos los niveles que integran la Universidad: órganos de gobierno, equipo de dirección, personal docente e investigador, personal de administración y servicios y estudiantes.

CSV:	bvlXDcmZNDohlYTso5JTtlxo9	Fecha:	16/01/2019 13:20:31	
Normativa:	Este documento es copia auténtica imprimible de un documento administrativo firmado electrónicamente y archivado por la Universidad Politécnica de Cartagena.			
Firmado Por:	Universidad Politécnica de Cartagena - Q8050013E			
Url Validación:	https://validador.upct.es/csv/bvlXDcmZNDohlYTso5JTtlxo9		Página:	

El Servicio de Prevención de Riesgos Laborales de la UPCT ha elaborado un “Manual de acogida al estudiante en materia de prevención de riesgos” que puedes encontrar en el Aula Virtual, y en el que encontraras instrucciones y recomendaciones acerca de cómo actuar de forma correcta, desde el punto de vista de la prevención (seguridad, ergonomía, etc.), cuando desarrolles cualquier tipo de actividad en la Universidad. También encontrarás recomendaciones sobre cómo proceder en caso de emergencia o que se produzca algún incidente.

En especial, cuando realices prácticas docentes en laboratorios, talleres o trabajo de campo, debes seguir todas las instrucciones del profesorado, que es la persona responsable de tu seguridad y salud durante su realización. Consúltale todas las dudas que te surjan y no pongas en riesgo tu seguridad ni la de tus compañeros.

5.4. Programa de teoría en inglés (unidades didácticas y temas)

U 1. Introduction to real-time systems

Lecture 0. Subject introduction

Lecture 1. Introduction to real-time systems

U 2. The C++ programming language

Lecture 2. Introduction to C++. C++ is not C (though they look alike)

Lecture 3. Object-oriented programming

Lecture 4. Standard C++ library: numeric, string, utilities, containers and iterators, algorithms, input/output

U 3. Concurrent programming

Lecture 5. Introduction to concurrent programming. Processes and threads. The scheduler of the operating system. POSIX

Lecture 6. Shared memory

U 4. Real-time systems scheduling

Lecture 7. Introduction to system scheduling

5.5. Objetivos del aprendizaje detallados por unidades didácticas

Unidad docente: Introducción a los sistemas de tiempo-real

- Objetivos de aprendizaje:**
- Describir las características fundamentales de los sistemas de tiempo-real y su clasificación
 - Describir los tipos de lenguajes que se pueden utilizar para diseñar sistemas de tiempo-real

Unidad docente: El lenguaje de programación C++

- Objetivos de aprendizaje:**
- Describir los tipos de datos de C++: primitivos, punteros, enumerados, clases y estructuras
 - Conocer y utilizar algunas características de C++: variables auto, inicialización uniforme, espacios de nombres, constantes, punteros
 - Conocer el mecanismo de excepciones de C++
 - Describir los principales contenedores definidos en la librería estándar
 - Conocer la relación contenedor-iterador-algoritmo implícita en la librería estándar y utilizarla para resolver algoritmos de tamaño medio
 - Definir y utilizar clases en C++. Gestión de memoria
 - Conocer el mecanismo de plantillas de C++

Tema: Programación concurrente

- Objetivos de aprendizaje:**
- Conocer las características y beneficios de los programas concurrentes
 - Definir y diferenciar los conceptos de proceso e hilo
 - Conocer los problemas asociados a la programación concurrente

- Conocer el concepto PCB y el ciclo de vida de los hilos
- Describir el funcionamiento del planificador del sistema operativo y los cambios de contexto
- Describir la gestión del tiempo de ejecución de los programas de un sistema informático que realiza un sistema operativo
- Describir los principales algoritmos de planificación de hilos
- Conocer el mecanismo de comunicación entre hilos mediante memoria compartida
- Describir los algoritmos para compartir memoria de forma segura mediante semáforos
- Utilizar la librería de creación de mutex
- Utilizar la librería de creación de variables de condición

Tema: Planificación de sistemas de tiempo-real

Objetivos de aprendizaje:

- Conocer la terminología utilizada en los sistemas de tiempo-real
- Describir el “problema de planificación” y los principales algoritmos de planificación utilizados
- Explicar el algoritmo “Ejecutivo Cíclico”, sus ventajas e inconvenientes
- Resolver un problema de planificación que utiliza el “Ejecutivo Cíclico”
- Explicar los algoritmos basados en prioridades fijas “Monotónico en Frecuencia” y “Monotónico en plazo”
- Describir y calcular el “factor de utilización” y el “instante crítico”
- Resolver problemas de planificación utilizando los algoritmos de planificación basados en prioridades fijas
- Calcular los tiempos de respuesta en sistemas basados en prioridades fijas
- Explicar los protocolos de evitación del problema de inversión de prioridad: “Herencia de Prioridad” y “Techo de Prioridad”
- Calcular los tiempos de respuesta en sistemas basados en prioridades fijas cuando existen recursos compartido

6. Metodología docente

6.1. Metodología docente*

Actividad*	Técnicas docentes	Trabajo del estudiante	Horas
Clase de teoría	Clase expositiva utilizando técnicas de aprendizaje cooperativo informal de corta duración. Resolución de dudas planteadas por los estudiantes. Orientación a los alumnos de los recursos documentales y multimedia disponibles de soporte al aprendizaje.	<u>Presencial</u> : Toma de apuntes. Planteamiento de dudas.	20
		<u>No presencial</u> : Estudio de la materia.	47
Clase de problemas	Se resolverán problemas tipo y se analizarán casos prácticos. Se enfatizará el trabajo en plantear métodos de resolución y no en los resultados. Se plantearán problemas y/o casos prácticos similares para que los alumnos lo vayan resolviendo, siendo guiados paso a paso por el profesor.	<u>Presencial</u> : Toma de apuntes y revisión con el compañero. Planteamiento de dudas.	10
		<u>No presencial</u> : Resolución de problemas y ejercicios como entregables semanales.	20
Clase de Prácticas. Sesiones de aula de informática	Las sesiones prácticas en el aula de informática son fundamentales para enlazar los contenidos teóricos y prácticos de forma directa. Mediante las sesiones de aula de informática se pretende que los alumnos apliquen los conocimientos de programación.	<u>Presencial</u> : Manejo del entorno de programación y resolución de problemas.	15
		<u>No presencial</u> : Lectura de la práctica con antelación al desarrollo de la misma y finalización en casa.	15
Actividades de evaluación formativa y sumativa	Se realizará una prueba parcial de carácter individual sobre los contenidos de la UD2 y diversos cuestionarios de Moodle. Se realizará una prueba escrita de carácter individual en convocatoria oficial para comprobar el grado de consecución de las competencias específicas.	<u>Presencial</u> : asistencia a la prueba escrita y realización de la misma.	5
		<u>No presencial</u> : Lectura de la práctica con antelación al desarrollo de la misma y finalización en casa.	3
			135

6.2. Resultados (4.5) / actividades formativas (6.1) (opcional)

Resultados del aprendizaje (4.5)							
Actividades formativas (6.1)	1	2	3	4	5	6	7
Clase de teoría	X		X	X		X	
Clase de problemas	X	X			X		X
Clase de prácticas en aula de informática	X				X		
Realización de test en Moodle	X	X			X		

7. Metodología de evaluación

7.1. Metodología de evaluación*

Actividad	Tipo		Sistema y criterios de evaluación*	Peso (%)	Resultados (4.5) evaluados
	Sumativa*	Formativa*			
(PE1) Prueba escrita individual parcial	X	X	Examen en aula informática. Opcional. La nota se guarda todo el curso académico. Recuperable.	20%	1,2,4,5
(PE2) Realización de tests	X	X	Tests de evaluación. Opcional. La nota se guarda solo en la convocatoria de evaluación continua.	15%	1,2,4,5
(PE3) Asistencia y realización de prácticas de informática		X	Mecanismos para evaluación del aprovechamiento en prácticas. Obligatoria su realización para aprobar la asignatura.	-	1,2,5
(PEI) Prueba escrita individual	X		1 pregunta teoría y 2 ejercicios de programación (C++ y concurrencia).	65% o 80%	1-7
<ul style="list-style-type: none"> - Para aprobar la asignatura hay que tener el APTO en prácticas (PE3), 1 falta máximo, y haber obtenido un mínimo de 4 sobre 10 en el examen final (PEI). - Después de la prueba (PEI) tendrá lugar el examen de recuperación del parcial (PE1) y las prácticas (PE3) en aula de informática. Si un alumno se presenta a la recuperación del parcial (PE1) se entiende que renuncian a la nota anterior. - En la primera convocatoria, de evaluación continua, el examen final (PEI) vale 6,5 puntos. En las siguientes dos convocatorias, de recuperación, el examen final (PEI) vale 8 puntos, ya que no se tiene en cuenta la nota de los tests (PE2). 					

Tal como prevé el artículo 5.4 del *Reglamento de las pruebas de evaluación de los títulos oficiales de grado y de máster con atribuciones profesionales* de la UPCT, el estudiante en el que se den las circunstancias especiales recogidas en el Reglamento, y previa solicitud justificada al Departamento y admitida por este, tendrá derecho a una prueba global de evaluación. Esto no le exime de realizar los trabajos obligatorios que estén recogidos en la guía docente de la asignatura.

7.2. Mecanismos de control y seguimiento (opcional)

La frecuencia y naturaleza de las preguntas realizadas por el alumnado, así como la intensidad de los correos electrónicos, la asistencia en los horarios establecidos para las tutorías y la participación en el aula virtual, son instrumentos medibles para determinar el avance formativo del estudiante, siempre que éste las utilice.

El primer trabajo de programación, presentado a mitad de curso, permite detectar posibles lagunas formativas al principio del proceso de aprendizaje.

De manera más específica, el seguimiento se realizará mediante las siguientes actividades:

- Los resultados de las pruebas de evaluación sumativas y formativas.
- La actitud de los alumnos en las prácticas.
- Estadísticas de uso del material colocado en el Aula Virtual.

- Cuestiones planteadas en clase puntualmente.
- El tiempo de resolución de los problemas planteados en el aula de informática.

8 Bibliografía y recursos

8.1. Bibliografía básica*

- *Professional C++*. Marc R. Gregoire, Nicholas Solter, Scott Kleper Ed.: Wiley Publishing, Inc., 2011. <http://site.ebrary.com/lib/univupct/Doc?id=10510437>
- *Programación Concurrente y Tiempo Real*. David Vallejo, Carlos González, Javier A. Albusac. 2016. Aul@ Virtual
- *Real Time Systems and Programming Languages*, Alan Burns, Andy Wellings. Addison Wesley, 2009.

8.2. Bibliografía complementaria

Muy recomendados:

- *Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14*, Scott Meyers. Ed. O'Reilly. 2014
- *Programming with POSIX Threads*. David R. Butenhof. Addison-Wesley Professional, 1997.

Para ampliar conocimientos:

- *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Hermann Kopetz. Springer, 2011.
- *Principles of Concurrent and Distributed Programming*, Mordechai Ben-Ari. Addison Wesley, 2006.
- *Programación Concurrente*, Palma Méndez, José T. y otros, Ediciones Paraninfo. S.A., 2003.

8.3. Recursos en red y otros recursos

Aula Virtual de la UPCT

Compilador online: <http://rextester.com/runcode>

Tutoriales y compiladores online de múltiples lenguajes de programación y desarrollo web:

<http://www.tutorialspoint.com/codingground.htm>

Referencia online C/C++: <http://en.cppreference.com/w/>

Programación de Sistemas en Tiempo Real: Planificación Temporal

		Semana															Otros	Periodo exámenes	TOTAL HORAS POR ACTIVIDAD	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15				
ACTIVIDADES PRESENCIALES	Convencionales	TEMA/ACTIVIDAD																		
		PARCIAL																		
		Clases teoría	3	3	2	1	1	1	1	1	1	1	1	1	1	1	1			
		Clases problemas			1	1	1	1	1	1		1		1	1	1				
		Aula de informatica				2	2	2	2		2	2	3							
		TOTAL CONVENCIONALES	3	3	3	4	4	4	4	2	3	4	4	2	2	2	1	0		
	No convencionales	Parcial								2										
		Realización de tests			0,5		0,5		0,5			0,5		0,5		0,5				
		PEI																3		
		TOTAL NO CONVENCIONALES	0	0	1	0	1	0	1	0	2	1	0	1	0	1	0	0	3	
ACTIVIDADES NO PRESENCIALES		Estudio	1	2	2	2	2	2	4	4		2	2	2	2	2	2	16		
		Problemas casa/semanales		2	2	2	2	2	2					2	2	2	2			
		Preparación de prácticas				2	2	2	2		2	2	3							
		TOTAL NO PRESENCIALES	1	4	4	6	6	6	8	4	2	4	5	4	4	4	4	0	16	
		TOTAL HORAS POR SEMANA	4	7	8	10	11	10	13	6	7	9	9	7	6	7	5	0	19	CARGA TOTAL
																			135	