



DATOS IDENTIFICATIVOS

Programación concurrente e distribuida

Asignatura	Programación concurrente e distribuida			
Código	V05G300V01641			
Titulación	Grado en Ingeniería de Tecnologías de Telecomunicación			
Descriptores	Creditos ECTS	Seleccione	Curso	Cuatrimestre
	6	OB	3	2c
Lengua	Castellano			
Impartición				
Departamento	Ingeniería telemática			
Coordinador/a	García Duque, Jorge			
Profesorado	García Duque, Jorge Pazos Arias, José Juan Ramos Cabrer, Manuel			
Correo-e	jgd@det.uvigo.es			
Web	http://fatic.uvigo.es			
Descripción general	El objetivo que se persigue con esta asignatura es que el alumno conozca los fundamentos de la sincronización y comunicación entre procesos tanto en sistemas centralizados como distribuidos.			

Competencias de titulación

Código	
A3	CG3 Conocimiento de materias básicas y tecnologías que capaciten al alumnado para el aprendizaje de nuevos métodos y tecnologías, así como que le dote de una gran versatilidad para adaptarse a nuevas situaciones.
A4	CG4 Capacidad para resolver problemas con iniciativa, para la toma de decisiones, la creatividad, y para comunicar y transmitir conocimientos, habilidades y destrezas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico de Telecomunicación.
A9	CG9 Capacidad para trabajar en un grupo multidisciplinar y en un entorno multilingüe y de comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.
A42	CE33/TEL7 Capacidad de programación de servicios y aplicaciones telemáticas, en red y distribuidas.

Competencias de materia

Resultados previstos en la materia	Resultados de Formación y Aprendizaje
Capacidad para diseñar y construir sistemas concurrentes y distribuidos.	A42
Comprensión de los principales conceptos teóricos de los sistemas concurrentes y distribuidos.	A3
Conocimiento de las principales herramientas y entornos para el desarrollo de sistemas concurrentes y distribuidos	A4
Capacidad para trabajar en grupo y presentar oralmente y por escrito los resultados de proyectos dentro del ámbito de la programación concurrente y distribuida	A9

Contenidos

Tema	
Introducción a la Programación Concurrente	☐ Conceptos de concurrencia, paralelismo y multitarea. ☐ Entrelazamiento de instrucciones atómicas. ☐ Grafos de precedencia.

El problema de la sección crítica	☐ Definición del problema. ☐ Espera Activa. ☐ Inanición ☐ Interbloqueo. ☐ El algoritmo de Decker. ☐ El algoritmo de Peterson
Herramientas de Sincronización	☐ Semáforos. ☐ El problema del productor-consumidor. ☐ El problema de los filósofos. ☐ Monitores. ☐ Variables de Condición. ☐ El problema de los lectores-escritores.
Gestión de Interbloqueo	☐ Introducción y definición de interbloqueo. ☐ Condiciones necesarias. ☐ Estrategias de Prevención. ☐ Estrategias de Evasión. ☐ Detección y Recuperación
Comunicación entre procesos	☐ Paso de Mensajes. ☐ Llamada a Procedimiento Remoto (RPC).
Programación Distribuida	☐ Introducción a los Sistemas Distribuidos. ☐ Exclusión Mutua Distribuida: - o Algoritmo Ricart-Agrawala. - o Algoritmos de paso de testigo. ☐ Consenso Distribuido: - o Fallos de parada. - o Fallos bizantinos.

Planificación

	Horas en clase	Horas fuera de clase	Horas totales
Talleres	5	30	35
Prácticas en aulas de informática	13	26	39
Sesión magistral	20	46	66
Pruebas de tipo test	1	0	1
Pruebas prácticas, de ejecución de tareas reales y/o simuladas.	1	0	1
Trabajos y proyectos	2	6	8

*Los datos que aparecen en la tabla de planificación son de carácter orientativo, considerando la heterogeneidad de alumnado

Metodologías

	Descripción
Talleres	Cada grupo de alumnos abordará el diseño e implementación de un proyecto software de complejidad media. Dicha tarea se realizará en diferentes pasos sucesivos, que serán discutidos y validados en cada una de las sesiones presenciales. Esta metodología de trabajo tiene como objetivo proporcionar una adecuada realimentación para, si es oportuno, mejorar las soluciones planteadas.
Prácticas en aulas de informática	Los alumnos resolverán bajo la supervisión del profesorado los problemas prácticos que se planteen en cada sesión de laboratorio.
Sesión magistral	Exposición de las ideas, conceptos, técnicas y algoritmos de cada lección del temario.

Atención personalizada

Metodologías	Descripción
Sesión magistral	Se dispensará atención personalizada de forma individual y presencial en el horario de tutorías que se hará público al inicio del curso. En las prácticas de laboratorio y talleres, la atención individualizada se articulará mediante el seguimiento del trabajo de cada alumno, monitorizando las soluciones parciales planteadas y reorientándolas si fuera preciso.
Talleres	Se dispensará atención personalizada de forma individual y presencial en el horario de tutorías que se hará público al inicio del curso. En las prácticas de laboratorio y talleres, la atención individualizada se articulará mediante el seguimiento del trabajo de cada alumno, monitorizando las soluciones parciales planteadas y reorientándolas si fuera preciso.
Prácticas en aulas de informática	Se dispensará atención personalizada de forma individual y presencial en el horario de tutorías que se hará público al inicio del curso. En las prácticas de laboratorio y talleres, la atención individualizada se articulará mediante el seguimiento del trabajo de cada alumno, monitorizando las soluciones parciales planteadas y reorientándolas si fuera preciso.

Evaluación		
	Descripción	Calificación
Pruebas de tipo test	Prueba de contenidos teóricos de cada uno de los temas expuestos en las sesiones magistrales.	60
Pruebas prácticas, de ejecución de tareas reales y/o simuladas.	Validación del trabajo realizado en cada una de las sesiones de laboratorio	18
Trabajos y proyectos	En la última sesión presencial del taller los alumnos entregarán y expondrán a sus compañeros el diseño y la solución planteados para el sistema software objetivo del proyecto. Dicha solución será expuesta a debate entre los alumnos y los profesores.	22

Otros comentarios sobre la Evaluación

La asignatura puede superarse mediante Evaluación Continua según los criterios que se indican más adelante, teniendo abierta la posibilidad de optar por la Evaluación No Continua en cualquier momento hasta el comienzo del examen final a celebrar el día fijado a tal efecto en el calendario oficial de la EET. Todos aquellos alumnos que opten por la evaluación continua se considerarán presentados si se evalúan de la parte del trabajo en Talleres.

Evaluación Continua:

La nota final resultará de la suma de las notas correspondientes a los tres componentes siguientes:

1. Cuatro pruebas de tipo Test para evaluar los contenidos impartidos en las clases magistrales. Cada prueba tendrá lugar en una de las sesiones magistrales, excepto la última que se realizará en una de las sesiones del Taller.

Puntuación: Hasta 1,5 puntos cada prueba.

2. Seis Pruebas Prácticas que se realizarán al finalizar cada una de las sesiones de laboratorio y que consistirán en la validación de los resultados obtenidos durante dicha sesión.

Puntuación: Hasta 0.3 puntos. cada prueba.

3. Presentación del Proyecto propuesto como trabajo en las sesiones del Taller.

Puntuación: Hasta 2.2 puntos.

Para aprobar la asignatura por Evaluación Continua se tendrán que dar las tres condiciones siguientes: (i) obtener una calificación igual o superior a 2 puntos en el conjunto de los tests.; (ii) calificación superior a 0 puntos en, al menos, cuatro de las seis pruebas prácticas; y (iii) asistir a todas las sesiones presenciales de taller y obtener más de 0 puntos en la presentación del proyecto.

Evaluación No Continua:

Mediante un examen sobre 10 puntos fijado en el calendario oficial de la EET.

Convocatoria de Fin de Curso:

Se registrá por lo indicado para la evaluación No Continua.

Fuentes de información

M. Ben-Ari, **Principles of Concurrent And Distributed Programming**, Second Edition,

George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, **Distributed Systems Concepts and Design**, Fifth Edition,

William Stallings, **Operating Systems: Internals and Design Principles, 6/E**, Seventh Edition,

Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, **Operating system concepts**, Eight Edition,

Lea, Douglas, **Programación concurrente en Java : principios y patrones de diseño**, Second Edition,

Recomendaciones

Asignaturas que se recomienda cursar simultáneamente

Arquitecturas y servicios telemáticos/V05G300V01645

Sistemas de información/V05G300V01644

Asignaturas que se recomienda haber cursado previamente

Informática: Arquitectura de ordenadores/V05G300V01103

Programación I/V05G300V01205

Programación II/V05G300V01302

Sistemas operativos/V05G300V01541
