

GUÍA DOCENTE DE

Programación Concurrente

Curso 2011-2012

TITULACION

Grado en Ingeniería de Computadores

GUIA DOCENTE DE LA ASIGNATURA

Programación Concurrente

Profesores

Nombre y apellidos: Micael Gallego Carrillo

Coordinador/a de la asignatura: Micael Gallego Carrillo

I.- Identificación de la asignatura

Tipo	Obligatoria
Materia	Lenguajes de Programación
Período de impartición	Primer semestre
Nº Créditos	6
Idioma en el que se imparte	Español
Departamento	Ciencias de la Computación
Asignaturas llave	-
Tasa de éxito	Este dato será incluido por el Vicerrectorado de Profesorado, Titulaciones, Ordenación Académica, Coordinación y Campus

II.- Presentación

El objetivo de la asignatura de Programación Concurrente es ofrecer al alumno una visión general de la programación concurrente y paralela. La programación concurrente es un paradigma de computación en el que los programas están diseñados como colecciones de procesos que interactúan y que pueden ser ejecutados en paralelo. En esta disciplina se estudian las técnicas utilizadas para la comunicación y sincronización de los procesos que se ejecutan en paralelo o concurrentemente.

La asignatura se estructura en dos partes: una parte con una mayor carga teórica que presenta las generalidades de la programación concurrente proporcionando al alumno una panorámica general de los conceptos y técnicas involucrados en la construcción de programas concurrentes; y otra parte más práctica que profundiza en el diseño e implementación de programas paralelos con herramientas profesionales.

La base de la programación concurrente es la programación secuencial, por ello se le presuponen al alumno conocimientos de programación adquiridos en la asignatura de Introducción a la programación (Primer curso) y en Estructuras de datos (Primer curso). Además, a lo largo del curso se realizan

ejercicios y prácticas utilizando mayoritariamente la plataforma Java, cuyo lenguaje se enmarca entre los lenguajes orientados a objetos. Por este motivo, se recomienda al alumno que tenga aprobada la asignatura de Programación Orientada a Objetos (Segundo curso), en la que se describe este paradigma y se realizan ejercicios prácticos en la plataforma Java.

La programación concurrente es una disciplina transversal en informática, ya que se encuentra presente en muchos tipos de sistemas. Es por ello que es recomendable que el alumno haya cursado satisfactoriamente las asignaturas de primer y segundo curso de la titulación. De esta forma, podrá tener una visión general sobre cómo la programación concurrente aparece en diferentes sistemas informáticos y diferentes niveles de abstracción, desde el nivel hardware, de sistema operativo y en el desarrollo de aplicaciones de alto nivel.

III.- Competencias

Competencias transversales	
G5	Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad
G9	Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática.
C6	Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos.
C8	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
C14	Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real.
C15	Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica.
Competencias específicas	
E2	E2. Capacidad de desarrollar procesadores específicos y sistemas empotrados, así como desarrollar y optimizar el software de dichos sistemas.

IV.- Contenido

IV. A. Temario de la asignatura

Bloque temático	Tema	Apartados
I.- Introducción	Tema 1. Introducción	¿Qué es la concurrencia? ¿Dónde se usa la concurrencia? ¿Cómo se usa la concurrencia? Conclusiones
	Tema 2. Sincronización de procesos con memoria compartida	Sincronización condicional con espera activa. Semáforos. Sincronización condicional con semáforos. Exclusión mutua con semáforos. Propiedades de corrección: Seguridad y Vida
II.- Desarrollo de Aplicaciones Concurrentes	Tema 3. Programación concurrente en Java	Introducción y fundamentos. Estructura de las aplicaciones concurrentes. Vida, rendimiento y pruebas
	Tema 4. Primitivas de sincronización en los lenguajes de programación	Paradigmas de programación. Lenguajes vs Librerías vs Sistemas Operativos. Memoria compartida. Paso de Mensajes.
	Tema 5. Aplicaciones concurrentes	Interfaces gráficas de usuario. Aplicaciones de red. Bases de datos. Algoritmos paralelos

IV. B. Actividades obligatorias (evaluables):

1. Lecturas

2. Prácticas

Se realizarán prácticas y ejercicios evaluables.

3. Laboratorios

--



4. Clínicos

5. Otras

V.- Tiempo de trabajo

Clases teóricas	22
Clases prácticas/de resolución de problemas, casos, etc.	0
Prácticas en laboratorios tecnológicos, clínicos, etc.	22
Realización de pruebas	4
Tutorías académicas	12
Actividades relacionadas: jornadas, seminarios, etc.	0
Preparación de clases teóricas	35
Preparación de clases prácticas/problemas/casos	35
Preparación de pruebas	20
Total de horas de trabajo del estudiante	150

VI.- Metodología y plan de trabajo

Clases teóricas

Periodo	Contenidos
Semana 1	Tema 1
Semanas 2-3	Tema 2
Semanas 4-9	Tema 3
Semanas 10-12	Temas 4 y 5



Prácticas/de resolución de problemas, casos, etc.

Periodo	Contenidos

Laboratorios

Periodo	Contenidos
Semanas 2-3	Tema 2
Semanas 4-9	Tema 3
Semanas 10-12	Temas 4 y 5

Seminarios y trabajos colectivos

Periodo	Contenidos

Tutorías académicas

Periodo	Contenidos
Semanas 1-12	Tutorías académicas

Otras actividades

Periodo	Contenidos

Pruebas

Fecha	Contenidos
Semana 4	Temas 1 y 2
Semana 13	Temas 3, 4 y 5



VII.- Métodos de evaluación

VII. A. Ponderación para la evaluación continua

Actividad evaluadora	Tipo		Ponderación	Periodo	Contenido
Prueba: <input checked="" type="checkbox"/> Preguntas cortas escritas 1	<input type="checkbox"/> Acumulativa				
	<input checked="" type="checkbox"/> Liberatoria Puntuación mínima (de 1 a 10): 5	<input checked="" type="checkbox"/> Reevaluable <input type="checkbox"/> No reevaluable	20%	Sem 4	Temas 1 y 2
Prueba: <input checked="" type="checkbox"/> Preguntas cortas escritas 2	<input type="checkbox"/> Acumulativa				
	<input checked="" type="checkbox"/> Liberatoria Puntuación mínima (de 1 a 10): 5	<input checked="" type="checkbox"/> Reevaluable <input type="checkbox"/> No reevaluable	40%	Sem 13	Temas 3, 4 y 5
Prácticas fuera del aula: <input checked="" type="checkbox"/> Resolución de problemas 1	<input type="checkbox"/> Acumulativa				
	<input checked="" type="checkbox"/> Liberatoria Puntuación mínima (de 1 a 10): 5	<input checked="" type="checkbox"/> Reevaluable <input type="checkbox"/> No reevaluable	10%	Sem 7	Práctica 1 Tema 3
Prácticas fuera del aula:	<input type="checkbox"/> Acumulativa				
	<input checked="" type="checkbox"/> Liberatoria Puntuación mínima	<input checked="" type="checkbox"/> Reevaluable	30%	Sem 13	Práctica 2 Temas 3

<input checked="" type="checkbox"/> Resolución de problemas 2	(de 1 a 10): 5	<input type="checkbox"/> No reevaluable			
Total			100%		

Se considerará que el estudiante ha adquirido las competencias evaluadas en cada **prueba escrita** si su calificación en dicha prueba es **mayor o igual a 5 sobre 10**. En caso contrario se considera que el estudiante no ha adquirido dichas competencias. Las competencias evaluadas en la prueba escrita no son reevaluables.

El material permitido para la realización de la prueba consistirá en el enunciado de la prueba, calculadora (no se admiten móviles ni relojes para este fin), útiles de escritura (excepto color rojo y lápiz). La prueba escrita es **individual**. El incumplimiento de estas normas se penalizará con la calificación de suspenso en la asignatura en la convocatoria evaluada donde se produzca dicho incumplimiento.

Se considerará que el estudiante ha adquirido las competencias evaluadas en cada prueba práctica si su calificación en dicha prueba es mayor o igual a 5 sobre 10 puntos. En caso contrario se considera que el estudiante no ha adquirido dichas competencias. Las competencias evaluadas en la prueba práctica no son reevaluables.

En caso de detectarse copias entre alumnos, el conjunto de la asignatura estará suspenso hasta la siguiente convocatoria. Todos los alumnos involucrados (tanto los que copian como los que son copiados) tendrán la asignatura suspensa hasta la siguiente convocatoria.

VII. B. Ponderación para la evaluación de alumnos a tiempo parcial

Para que un alumno pueda optar a esta evaluación, tendrá que obtener la "Dispensa Académica" para la asignatura, que habrá solicitado al Decano o Director/a del Centro que imparte su titulación.

La "Dispensa Académica" no excluye de la evaluación continua. Dicha evaluación se acomodará por el profesor, asistido por el coordinador de grado, estableciéndose la adaptación curricular según las características de cada caso concreto.

VII. C. Revisión de las pruebas de evaluación.

La revisión de las pruebas de evaluación se llevará a cabo en clase o en el horario de tutorías de la asignatura.

VIII.- Recursos y materiales didácticos

General

Título	Java Concurrency in Practice
Autor	Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea
Editorial	Addison Wesley Professional
Título	Programación concurrente
Autor	Sánchez Figueroa, Fernando ; Garrido Carrera, M ^a Del Carmen ; Palma Méndez, José Tomas ; Quesada Arencibia, Alexis
Editorial	THOMSON PARANINFO,S.A

Complementaria

Título	Patterns for Parallel Programming
Autor	Timothy G. Mattson, Beverly A. Sanders and Berna L. Massingill
Editorial	Addison-Wesley Professional
Título	Principles of concurrent and distributed programming
Autor	Ben-Ari, Mordechai
Editorial	Addison-Wesley

Direcciones web de interés

<p>Wikipedia</p> <ul style="list-style-type: none"> • Concurrent Programming: http://en.wikipedia.org/wiki/Concurrent_programming • Threads: http://en.wikipedia.org/wiki/Thread_(computer_science) • Parallel Computing: http://en.wikipedia.org/wiki/Parallel_computing
<p>Open Directory Project</p> <ul style="list-style-type: none"> • Concurrent Languages: http://www.dmoz.org/Computers/Programming/Languages/Concurrent/



- Threads: <http://www.dmoz.org/Computers/Programming/Threads/>
- Parallel Computing: http://www.dmoz.org/Computers/Parallel_Computing/

IX.- Profesorado

Nombre y apellidos	Micael Gallego Carrillo
Horario de tutorías académicas	Se publicarán más tarde
Correo electrónico	micael.gallego@urjc.es
Departamento/área de conocimiento	Ciencias de la Computación
Categoría	Profesor Contratado Doctor
Titulación Académica	Doctor en Ciencias de la Computación por la Universidad Rey Juan Carlos
Experiencia Docente (Indicar la antigüedad en el área y en la asignatura. Incluir tramos de docencia.)	<p>Profesor de la Universidad Rey Juan Carlos desde Febrero de 2007. Ha impartido asignaturas de grado y postgrado en las temáticas de: Paradigmas de programación (Funcional, Concurrente, Orientado a Objetos, Orientado a Eventos), Procesadores de Lenguajes, Seguridad Informática, Ingeniería del Software, Software avanzado, Diseño de Algoritmos e Investigación Operativa (Optimización).</p> <p>Ha obtenido en el año 2011 la evaluación positiva del programa Docencia de la URJC. Dentro de sus intereses se encuentra la mejora en la docencia, habiendo participado en dos proyectos de innovación educativa, la elaboración de dos libros docentes y en el desarrollo de herramientas educativas (entre las que destaca EclipseGavab: http://code.sidelab.es/projects/eclipsegavab).</p>
Experiencia profesional (Indicar la actividad profesional y la antigüedad en la misma)	<p>Pertenece al grupo de investigación Gavab (http://www.gavab.es) desde 2005 y su investigación se centra en el diseño, implementación y validación de algoritmos de resolución aproximada de problemas de optimización, usando las novedosas técnicas metaheurísticas. Codirige el laboratorio de desarrollo software Sidelab (http://www.sidelab.es), que actúa como punto de encuentro de desarrolladores y entusiastas de la programación con la impartición de seminarios, desarrollo de herramientas con licencias libres, desarrollo de proyectos de fin de grado y máster, etc.</p>



	<p>De forma complementaria, desde 2006 colabora activamente con empresas del sector de las tecnologías de la información prestando servicios de consultoría y dirección de proyectos. Cabe destacar la colaboración entre los años 2007 y 2010 con la empresa Solaiemes (http://www.solaiemes.com) en el desarrollo de sistemas de emisión y gestión de video en tiempo real en Internet y teléfonos móviles. Actualmente es CIO (<i>Chief Information Officer</i>) de TS Company (http://www.tscompany.es), empresa tecnológica especializada en software relacionado con la actividad física y el deporte.</p>
--	---