



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid

Concurrencia

Guía de Aprendizaje – Información al estudiante

Datos Descriptivos

Asignatura	Concurrencia
Materia	Programación
Departamento responsable	Lenguajes, Sistemas Informáticos e Ingeniería de Software
Créditos ECTS	3
Carácter	Obligatorio
Titulación	Graduado en Ingeniería Informática
Curso	2º
Especialidad	No aplica

Curso académico	2012-2013
Semestre en que se imparte	3S,4S
Semestre principal	4S
Idioma en que se imparte	Castellano
Página Web	http://babel.ls.fi.upm.es/teaching/concurrencia



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid

Profesorado

NOMBRE Y APELLIDO	DESPACHO	Correo electrónico
Julio Mariño Carballo (Coord.)	2308	jmarino@fi.upm.es
Angel Herranz Nieva	2309	aherranz@fi.upm.es
Lars-Åke Fredlund	2309	lfredlund@fi.upm.es

Conocimientos previos requeridos para poder seguir con normalidad la asignatura

Asignaturas superadas	<ul style="list-style-type: none">• Programación I• Programación II• Matemática Discreta I• Lógica• Lenguajes Formales, Autómatas y Computabilidad
Otros resultados de aprendizaje necesarios	



Objetivos de Aprendizaje

COMPETENCIAS ASIGNADAS A LA ASIGNATURA Y SU NIVEL DE ADQUISICIÓN		
Código	Competencia	Nivel
CE-2	Formalización y especificación de problemas reales cuya solución requiere el uso de la informática.	3
CE-24	Elegir y usar los lenguajes de programación adecuados al tipo de aplicación a desarrollar.	3
CE-25	Concebir y diseñar la arquitectura de un sistema software.	2
CE-29	Diseñar, desarrollar, y evaluar la seguridad de los sistemas, aplicaciones, servicios informáticos y sistemas operativos sobre los que se ejecutan, así como de la información que proporcionan.	3
CE-40	Comprender el concepto esencial de proceso en cuanto a su relación con la informática, especialmente la ejecución de los programas y la operación del sistema.	3

COMPETENCIAS TRANSVERSALES ASIGNADAS A LA ASIGNATURA Y SU NIVEL DE ADQUISICIÓN		
Código	Competencia	Nivel
CG#1/21	Capacidad de resolución de problemas aplicando conocimientos de matemáticas, ciencias e ingeniería.	2
CG#2/CE45	Capacidad para el aprendizaje autónomo y la actualización de conocimientos, y reconocimiento de su necesidad en el área de la informática.	2

LEYENDA: Nivel de adquisición 1:
Nivel de adquisición 2:
Nivel de adquisición 3:



RESULTADOS DE APRENDIZAJE DE LA ASIGNATURA			
Código	Resultado de aprendizaje	Competencias asociadas	Nivel de adquisición
RA1	Reconocer la concurrencia inherente a un sistema SW	CE-25, CE-40	2,3
RA2	Especificar propiedades relativas a la ejecución concurrente de un sistema SW	CE-2, CE-29, CE-40, CG#1/21	3
RA3	Desarrollar aplicaciones concurrentes en Java	CE-24, CG#2/CE45	3
RA-4	Analizar propiedades y riesgos atribuibles a la ejecución concurrente de un sistema SW	CE-29, CE-40, CG#1/21	3



Sistema de evaluación de la asignatura

INDICADORES DE LOGRO		
Ref.	Indicador	Relacionado con RA
I1	Diseñar una arquitectura de procesos e interacciones a partir de las acciones contenidas en la especificación de requisitos de un sistema SW	RA-1
I2	Formalizar propiedades de seguridad mediante un lenguaje de recursos compartidos	RA-2
I3	Formalizar propiedades dinámicas mediante un lenguaje de trazas	RA-2
I4	Demostrar conocimiento de los mecanismos de concurrencia del lenguaje	RA-3
I5	Caso práctico: traducir a Java una especificación de recurso compartido, siguiendo patrones de código	RA-3
I6	Encontrar un error de seguridad o vivacidad en un sistema concurrente dado	RA-4

INDICADORES DE LOGRO DE COMPETENCIAS TRANSVERSALES		
Ref.	Indicador	Relacionado con CT
I2	Formalizar propiedades de seguridad mediante un lenguaje de recursos compartidos	CG#1/21
I3	Formalizar propiedades dinámicas mediante un lenguaje de trazas	CG#1/21
I4	Demostrar conocimiento de los mecanismos de concurrencia del lenguaje	CG#2/CE45
I5	Caso práctico: traducir a Java una especificación de recurso compartido, siguiendo patrones de código	CG#2/CE45
I6	Encontrar un error de seguridad o vivacidad en un sistema concurrente dado	CG#1/21



EVALUACION SUMATIVA			
Breve descripción de las actividades evaluables	Momento	Lugar	Peso en la calif.
Ejercicio corto: creación de procesos en Java	Semana 2	telemática	1%
Ejercicio corto: provocar una situación de carrera	Semana 2	telemática	1%
Ejercicio corto: exclusión mutua con espera activa	Semana 3	telemática	1%
Ejercicio corto: reparar carrera con semáforos	Semana 4	telemática	1%
Ejercicio corto: almacén de un dato con semáforos	Semana 5	telemática	1%
Ejercicio corto: almacén de varios datos con semáforos	Semana 6	telemática	1%
Ejercicio corto: especificación de recursos sencillos	Semana 7	telemática	1%
Test de preguntas de opción múltiple para los temas 1-2	Semana 8	aula	25%
Ejercicio corto: MultiBuffer con métodos synchronized	Semana 9	telemática	1%
Ejercicio corto: MultiBuffer con cerrojos	Semana 10	telemática	1%
Práctica por parejas: implementación de recursos compartidos en Java con cerrojos y variables condition	Semanas 11-13	telemática	20%
Ejercicio corto: MultiBuffer con JCSP	Semana 14	individual	1%
Práctica por parejas: implementación de recursos compartidos en Java con la librería JCSP	Semanas 13-15	telemática	20%
Test de preguntas de opción múltiple para el tema 3	Semana 15	aula	25%
Total:			100%



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid

EVALUACION SUMATIVA DE COMPETENCIAS TRANSVERSALES				
CÓD. COMPETENCIA TRANSVERSAL	Breve descripción de las actividades evaluables	Momento	Lugar	Peso en la calif.
CG#1/21	Test de preguntas de opción múltiple para los temas 1-2	Semana 8	aula	40%
CG#1/21	Práctica por parejas: implementación de recursos compartidos en Java con cerrojos y variables condition	Semanas 11-13	telemática	10%
CG#1/21	Práctica por parejas: implementación de recursos compartidos en Java con la librería JCSP	Semanas 13-15	telemática	10%
CG#1/21	Test de preguntas de opción múltiple para el tema 3	Semana 15	aula	40%
CG#2/CE45	Ejercicios cortos (entregas 1 a 10)	Semana 2	telemática	40%
CG#2/CE45	Práctica por parejas: implementación de recursos compartidos en Java con cerrojos y variables condition	Semanas 11-13	telemática	30%
CG#2/CE45	Práctica por parejas: implementación de recursos compartidos en Java con la librería JCSP	Semanas 13-15	telemática	30%



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid



POLITÉCNICA



UNIVERSIDAD POLITÉCNICA DE MADRID
FACULTAD DE INFORMÁTICA
Campus de Montegancedo
Boadilla del Monte. 28660 Madrid



La asignatura Concurrencia tiene dos partes evaluables, una parte teórica y una parte práctica. Para aprobar la asignatura el estudiante deberá superar ambas partes. Además, el estudiante debe tener en cuenta que la asistencia a clase es absolutamente fundamental para superar la asignatura y el profesorado realizará un seguimiento de dicha asistencia.

Teoría

La parte teórica la componen dos tests. El primero tratará los conceptos de los temas 1 y 2 y se realizará alrededor de la semana 8. El segundo tratará los conceptos del tema 3 y se realizará en la semana 15. Cada test recibirá una calificación sobre 10 (nt1 y nt2).

La nota de la parte teórica (nt) se calcula entonces como la media de nt1 y nt2. Se considerará superada la parte teórica cuando nt1 y nt2 sean mayores o iguales a 4 y cuando nt sea mayor o igual a 5.

Práctica

La parte práctica la componen un conjunto de 10 ejercicios cortos cuyo objetivo es garantizar el seguimiento continuo de la asignatura por parte del alumno, y dos prácticas a realizar por parejas. La entrega de todos los ejercicios es de carácter obligatorio, y se califican sobre 10 puntos, obteniéndose ne, np1 y np2.

Se considerará superada la parte práctica cuando ne, np1 y np2 sean mayores o iguales a 3 y cuando np $((ne+2np1+2np2)/5)$ sea mayor o igual a 5.

Total

La asignatura se considerará superada cuando se superen la parte teórica y la parte práctica. La calificación final (nf) se calcula como la media de nt y np.

Superada cualquiera de las partes (nt1, nt2, ne, np1, np2) la calificación se guardará (sólo) hasta la convocatoria de julio del curso en cuestión.



CRITERIOS DE CALIFICACIÓN

Evaluación en modo “sólo examen final”

Para aquellos alumnos que soliciten evaluación de la asignatura en modo “sólo examen final” en los plazos establecidos por la Universidad, se realizará un examen final en el que se evaluarán todos los bloques arriba mencionados, y que constará de:

- 2 tests equivalentes a nt1 y nt2
- Un ejercicio práctico de implementación ante ordenador, que cubra las pruebas prácticas de la asignatura (ne, np1, np2).

Actuación ante copias y otros comportamientos fraudulentos

En la asignatura Concurrencia la realización de los ejercicios prácticos es fundamental para la adquisición de los resultados de aprendizaje (como en toda asignatura relacionada con la construcción de SW).

Por ello, se perseguirá todo tipo de comportamiento fraudulento (como la copia de prácticas) y los alumnos implicados perderán las convocatorias del curso. El asunto será notificado a Jefatura de Estudios y a los tutores curriculares de los alumnos.

Se prevé el uso de exámenes de la práctica a una selección de grupos como medio de lucha contra las copias.



Contenidos y Actividades de Aprendizaje

1. Introducción: concurrencia e interacción

- 1.1. Conceptos fundamentales de concurrencia (I4, I6)
- 1.2. Manejo básico de procesos en Java (I5)
- 1.3. Sincronización de bajo nivel (I6)

2. Metodología: recursos compartidos

- 2.1. Notación formal para especificar recursos compartidos (I1, I2)
- 2.2. Semántica de trazas de un sistema basado en recursos compartidos (I3)

3. Implementación de recursos compartidos en Java

- 3.1. Memoria compartida con métodos *synchronized* (I5)
- 3.2. Memoria compartida con cerrojos y variables *condition* (I5)
- 3.3. Paso de mensajes con la librería JCSP (I5)



Breve descripción de las modalidades organizativas utilizadas y de los métodos de enseñanza empleados

BREVE DESCRIPCIÓN DE LAS MODALIDADES ORGANIZATIVAS UTILIZADAS Y METODOS DE ENSEÑANZA EMPLEADOS	
CLASES DE TEORIA	...
CLASES DE PROBLEMAS	...
PRÁCTICAS	...
TRABAJOS AUTONOMOS	...
TRABAJOS EN GRUPO	...
TUTORÍAS	...

Recursos didácticos

RECURSOS DIDÁCTICOS	
BIBLIOGRAFÍA	Artículos varios
	Apuntes del profesorado
	"The Theory and Practice of Concurrency". A.W. Roscoe. Prentice-Hall, 1997
	Concurrent and Real-Time Programming in Java. Andy Wellings. Wiley, 2004
	Concurrent Programming in Java: Design Principles and Patterns. Doug Lea. Addison-Wesley, 1999
RECURSOS WEB	Página web de la asignatura (http://babel.ls.fi.upm.es/teching/concurrencia)
	Foro de la asignatura (foro-cc@babel.ls.fi.upm.es)
EQUIPAMIENTO	Aula

Cronograma de trabajo de la asignatura

Semana	Actividades en Aula	Actividades en Laboratorio	Trabajo Individual	Trabajo en Grupo	Actividades de Evaluación	Otros
Semana 1 (5 horas)	Presentación de la asignatura (1h)	.	Lectura: Concepts and Notations for Concurrent Programming (4 horas)	.	.	.
Semana 2 (5 horas)	Preguntas y respuestas sobre conceptos básicos de concurrencia (1h).	.	Lectura: Tutorial de concurrencia de java.sun.com (1h) Ejercicio: Creación de procesos en Java (1h)	.	.	.
	Puesta en común (1h).	.	Ejercicio: Provocar una carrera (1h)	.	.	.
Semana 3 (5 horas)	Exclusión mutua; Algoritmos basados en espera activa (1h).	.	Ejercicio: implementación de algoritmos de exclusión mutua basados en espera activa (3h).	.	.	.
	Problemas de concurrencia en algoritmos de espera activa(1h).
Semana 4 (3 horas)	Exclusión mutua con semáforos (1h).	.	Ejercicio: reparar una condición de carrera con semáforos. (1h)	.	.	.
	Puesta en común (1h).
Semana 5 (5 horas)	Sincronización condicional (1h)	.	Ejercicio: productor-dato-consumidor con semáforos (3h).	.	.	.
	Puesta en común: análisis de trazas y estados (1h).
Semana 6 (5 horas)	Sincronización de alto nivel: recursos compartidos (1h).	.	Ejercicio: productor-almacén-consumidor con semáforos (3h).	.	.	.
	Recursos compartidos: semántica de trazas (1h).
Semana 7 (5 horas)	Recursos compartidos y problemas de exclusión parcial (lectores/escritores) (1h).	.	Ejercicio: especificación de recursos sencillos (3h).	.	.	.

Semana	Actividades en Aula	Actividades en Laboratorio	Trabajo Individual	Trabajo en Grupo	Actividades de Evaluación	Otros
	Puesta en comun (1h).
Semana 8 (5 horas)	Repaso (1h).
	Repaso (1h).
		.	Estudio individual (2h).	.	Prueba objetiva (1h)	.
Semana 9 (5 horas)	Concurrencia mediante memoria compartida en Java: métodos <i>synchronized</i> (1h).	.	Ejercicio: MultiBuffer con métodos <i>synchronized</i> (3h)	.	.	.
	Puesta en común (1h).
Semana 10 (5 horas)	Concurrencia mediante memoria compartida en Java: cerrojos y variables <i>condition</i> (1h).	.	Ejercicio: MultiBuffer con cerrojos (3h).	.	.	.
	Puesta en común (1h).
Semana 11 (5 horas)	Repaso, ejercicios en clase (1h).
	Explicación de la práctica (1h).	.	Desarrollo de caso práctico (3 horas).	.	.	.
Semana 12 (6 horas)	Tutoría colectiva (1h).
	Tutoría colectiva (1h).	.	Desarrollo de caso práctico (4 horas).	.	.	.
Semana 13 (5 horas)	Concurrencia mediante paso de mensajes: conceptos básicos (1h).	.	Lectura: Andrews + Schneider (1h)	.	.	.
	Concurrencia mediante paso de mensajes en Java: la librería JCSP (1h).	.	Desarrollo de caso práctico (2h)	.	.	.
Semana 14 (6)	Tutoría colectiva (1h).	.	Ejercicio: MultiBuffer en JCSP (2h)	.	.	.

Semana	Actividades en Aula	Actividades en Laboratorio	Trabajo Individual	Trabajo en Grupo	Actividades de Evaluación	Otros
horas)	Tutoría colectiva (1h).	.	·Desarrollo de caso práctico (2h)	.	.	.
Semana 15 (5 horas)	Repaso + tutoría colectiva (1h).	.	Estudio individual (2h)	.	.	.
	Repaso + tutoría colectiva (1h).	.	.	.	·Prueba objetiva (1h)	.
Semana 16 (5 horas)	Afianzamiento de prácticas (1h)
	Afianzamiento de prácticas (1h).	.				.
Semana 17 (período de exámenes, ??? horas)

Nota: Para cada actividad se especifica la dedicación en horas que implica para el alumno.