



DATOS IDENTIFICATIVOS

Concurrencia y distribución

Asignatura	Concurrencia y distribución			
Código	O06G151V01308			
Titulación	Grado en Ingeniería Informática			
Descriptores	Creditos ECTS	Seleccione	Curso	Cuatrimestre
	6	OB	3	2c
Lengua	#EnglishFriendly			
Impartición	Castellano Gallego			
Departamento	Informática			
Coordinador/a	Formella , Arno			
Profesorado	Formella , Arno			
Correo-e	formella@uvigo.es			
Web	http://formella.webs.uvigo.es/doc/cdg22			

Descripción general El contenido forma la base necesaria para comprender el funcionamiento de aplicaciones concurrentes y/o distribuidas, la evaluación de algoritmos concurrentes, la descripción de datos y de información en entornos distribuidos, el funcionamiento de procesadores modernos, y las características especiales de la programación con procesos/hilos incluso en forma distribuida.

Las clases se darán principalmente en castellano. El/la estudiante puede elegir si trabaja en gallego, castellano, alemán, portugués y/o en inglés. Cierta información adicional (como por ejemplo manuales e información complementaria) se dará en inglés.

Asignatura del programa English Friendly: Los/as estudiantes internacionales podrán solicitar al profesorado: a) materiales y referencias bibliográficas para el seguimiento de la materia en inglés, b) atender las tutorías en inglés, c) pruebas y evaluaciones en inglés.

Competencias

Código	
A1	Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.
A2	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.
A3	Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética.
A5	Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.
B5	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, de acuerdo con los conocimientos adquiridos.
B6	Capacidad para concebir y desarrollar sistemas o arquitecturas informáticas centralizadas o distribuidas integrando hardware, software y redes de acuerdo con los conocimientos adquiridos.
B9	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.
C14	Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados
C15	Capacidad de conocer, comprender y evaluar la estructura y arquitectura de los computadores, así como los componentes básicos que los conforman

C16	Conocimiento de las características, funcionalidades y estructura de los Sistemas Operativos y diseñar e implementar aplicaciones basadas en sus servicios
C17	Conocimiento y aplicación de las características, funcionalidades y estructura de los Sistemas Distribuidos, las Redes de Computadores e Internet y diseñar e implementar aplicaciones basadas en ellas
C20	Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real
C28	Capacidad de identificar y analizar problemas y diseñar, desarrollar, implementar, verificar y documentar soluciones software sobre la base de un conocimiento adecuado de las teorías, modelos y técnicas actuales
D4	Capacidad de análisis, síntesis y evaluación
D5	Capacidad de organización y planificación
D6	Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales
D7	Capacidad de buscar, relacionar y estructurar información proveniente de diversas fuentes y de integrar ideas y conocimientos.
D8	Capacidad de trabajar en situaciones de falta de información y/o bajo presión
D10	Capacidad de relación interpersonal.
D11	Razonamiento crítico
D14	Tener motivación por la calidad y la mejora continua

Resultados de aprendizaje

Resultados previstos en la materia	Resultados de Formación y Aprendizaje			
RA1: Conocer los fundamentos teóricos de los sistemas concurrentes y distribuidos.	A1	B5	C14	D4
	A2	B6	C15	D5
	A3	B9	C16	D6
	A5		C17	D7
			C20	D8
			C28	D11
				D14
RA2: Conocer sistemas y entornos con concurrencia y distribución.	A1	B5	C14	D4
	A2	B6	C15	D5
	A3	B9	C16	D6
	A5		C17	D7
			C20	D8
			C28	D11
				D14
RA3: Conocer el proceso de generación de aplicaciones para sistemas concurrentes y distribuidos.	A1	B5	C14	D4
	A2	B6	C15	D5
	A3	B9	C16	D6
	A5		C17	D7
			C20	D8
			C28	D10
				D11
				D14
RA4: Conocer las herramientas y sus propiedades en uso para generar código para sistemas concurrentes y distribuidos.	A1	B5	C14	D4
	A2	B6	C15	D6
	A3	B9	C16	D7
	A5		C17	D8
			C20	D10
			C28	D11
				D14

Contenidos

Tema	
Sistemas concurrentes y distribuidos	Concepto de la programación concurrente y distribuida Introducción a los sistemas concurrentes y distribuidos Diseño Software Concurrente Herramientas para el desarrollo de aplicaciones concurrentes y distribuidas
Procesos	Concepto de procesos Atomicidad y exclusión mutua Concurrencia transaccional Estado distribuido
Sincronización y comunicación	Sincronización y comunicación en sistemas concurrentes y distribuidos Sincronización y comunicación a nivel bajo y alto Propiedades de sistemas concurrentes y distribuidos

Planificación			
	Horas en clase	Horas fuera de clase	Horas totales
Actividades introductorias	0.5	0	0.5
Lección magistral	18	10	28
Estudio previo	0	18	18
Prácticas de laboratorio	24	26	50
Resolución de problemas	1.5	19.5	21
Presentación	0	2	2
Foros de discusión	2.5	0	2.5
Resolución de problemas y/o ejercicios	1	0	1
Examen de preguntas de desarrollo	2	0	2
Informe de prácticas, prácticum y prácticas externas	0	12	12
Práctica de laboratorio	1	0	1
Resolución de problemas y/o ejercicios	2	10	12

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

Metodologías	
	Descripción
Actividades introductorias	Presentación de la asignatura y aclaración de todos los aspectos relacionados con la guía docente. Planificación temporal de las actividades presenciales. Introducción de las herramientas de control y evaluación. Recomendaciones específicas para lograr los objetivos de la asignatura.
Lección magistral	Exposición de los contenidos teóricos de la asignatura. Presentación de ejemplos y casos específicos. Controles de estudios y lecturas previos. Controles del avance de la adquisición de conocimiento por parte del estudiante. Interacción con/entre los estudiantes mediante actividades específicas.
Estudio previo	Lectura de documentos relacionados con el contenido de la asignatura. Análisis y diseño de las tareas de los ejercicios en el laboratorio.
Prácticas de laboratorio	Implementación y depuración de los ejercicios planteados como tareas de programación. Realización de pruebas de funcionamiento y/o rendimiento de aplicaciones concurrentes y distribuidas con un análisis crítico de las observaciones.
Resolución de problemas	Desarrollo de propuestas de resolución de problemas. Comprobaciones de corrección y análisis de rendimiento. Implementación de soluciones alternativas. Análisis crítico de soluciones propuestas.
Presentación	Exposiciones breves de los hitos alcanzados en las tareas de programación y ejercicios relacionados.
Foros de discusión	Discusión y debate en grupos pequeños o grandes sobre temas concretos de la asignatura y posibles soluciones a problemas presentados con la presencia del profesorado.

Atención personalizada	
Metodologías	Descripción
Actividades introductorias	Presentación de la organización, de la evaluación, y de las competencias.
Lección magistral	El profesor muestra de manera resumida y estructurada la información y el conocimiento intrínseco del contenido específico de la materia interrelacionando las diferentes partes y enlazando los conceptos entre sí, con la bibliografía y con las prácticas.
Presentación	El/la estudiante expone al profesor y/o a un grupo de estudiantes el diseño de su solución y los resultados obtenidos.
Prácticas de laboratorio	El/la estudiante trabaja en las tareas según boletín de prácticas publicado a lo largo del curso para tal fin aprovechando la presencia del profesor.
Foros de discusión	El o la estudiante realiza preguntas que considera oportunas relacionadas con el temario o el proceso de aprendizaje. Se intenta incentivar debate activa entre los participantes.
Pruebas	Descripción
Resolución de problemas y/o ejercicios	El/la estudiante contesta a un conjunto de preguntas cortas por escrito.

Examen de preguntas de desarrollo	El/la estudiante contesta a un conjunto de preguntas largas en un contexto específico con razonamiento.
Informe de prácticas, prácticum y prácticas externas	El/la estudiante elabora informes documentando las decisiones tomadas y los resultados obtenidos incluyendo razonamiento crítico.
Práctica de laboratorio	El/la estudiante muestra que sus implementaciones de las tareas de programación cumplan con los requisitos especificados.
Resolución de problemas y/o ejercicios	El/la estudiante muestra de manera más el menos formal que las soluciones tengan las propiedades requeridas.

Evaluación			
	Descripción	Calificación	Resultados de Formación y Aprendizaje
Presentación	(P5) Elaboración de algoritmos o aplicaciones e as súas análises con certo nivel de formalismo para comprobar a corrección e estudar o rendemento. Avaliación cunha puntuación de 1-10, participación optativa e voluntaria. (RA1, RA2, RA3, RA4)	5	B5 C14 D4 B6 C15 D5 B9 C16 D6 C17 D7 C20 D8 C28 D11 D14
Resolución de problemas y/o ejercicios	(P1) Conjunto de preguntas cortas para el control de la realización de actividades, tareas, y estudios. Media de las pruebas realizadas con puntuación de 1-10. (RA1, RA2, RA3, RA4)	10	A1 B5 C14 D4 A2 B6 C15 D5 B9 C16 D6 C17 D7 C20 D8 C28 D10 D11 D14
Examen de preguntas de desarrollo	(P2) Conjunto de preguntas largas que relacionan los diferentes apartados de los contenidos y miden el nivel de adquisición de las competencias de la materia. Prueba con puntuación de 1-10, mínimo requerido: 4. (RA1, RA2, RA3, RA4)	40	A1 B5 C14 D4 A2 B6 C15 D6 B9 C16 D7 C17 D8 C20 D11 C28 D14
Informe de prácticas, prácticum y prácticas externas	(P3) Elaboración de informes (según una guía) que recogen los principales desarrollos y resultados obtenidos por el/la estudiante. Parte de dichos informes se elaborará en pequeños grupos. Media de evaluaciones de las actividades con puntuaciones de 1-10. (RA1, RA2, RA3, RA4)	25	A3 B5 C14 D4 A5 B6 C15 D5 B9 C16 D6 C17 D7 C20 D8 C28 D10 D11 D14
Práctica de laboratorio	(P4) Demostración de los desarrollos e implementación de las tareas de programación y experimentos de estudio. Media de evaluaciones de las actividades con puntuaciones de 1-10., mínimo requerido: 4. (RA1, RA2, RA3, RA4)	25	A3 B5 C14 D4 A5 B6 C15 D5 B9 C16 D6 C17 D7 C20 D8 C28 D10 D11 D14

Otros comentarios sobre la Evaluación

Se entiende que un estudiante que no participa en al menos el 80% de las actividades presenciales no asiste y, por lo tanto, se somete a la forma de evaluación para no asistir. La decisión de no ser asistente la toma el estudiante al comienzo del curso o el profesorado cuando está claro que ya no es posible alcanzar el 80% de asistencia en las actividades presenciales.

Crterios de evaluación para los asistentes 1ª edición del acta:

Evaluación de la teoría: finalización de diferentes actividades (por ejemplo, lectura de bibliografía, elaboración de esquemas, estudio de manuales, etc.) y verificación mediante pruebas cortas escritas a lo largo del curso (incluso inesperadamente) que recopilarán contenidos teóricos y prácticas correspondientes a la asignatura impartida durante las clases presenciales y en los estudios autónomos (P1). Realización de un examen escrito final (P2) con diferentes preguntas.

Evaluación de prácticas: evalúa la realización de los ejercicios de elaboración propuestos (P4), la elaboración de la documentación y de los informes relacionados (P3). Realización de un trabajo opcional hacia el final de las clases de

contacto (P5).

Para los asistentes, el espíritu del cálculo de la nota final de la materia es el siguiente: hay una combinación de pruebas teóricas y prácticas a lo largo del curso y, a su fin, para evaluar las competencias adquiridas. Los buenos resultados en una parte pueden compensar los resultados no tan buenos en otra parte, sin embargo, se debe alcanzar un mínimo en las dos secciones más relevantes (P2 y P4).

La calificación final se obtiene de la siguiente manera, asumiendo que cada parte (P1-P5) se evalúa con una escala de 0-10:

El estudiante suspende si la calificación de P2 o P4 es menor de 4 (la nota de la parte más alta que motiva el suspenso aparecerá en las actas). La asignatura se aprueba si se cumple $\min(10, 0.1 * P1 + 0.4 * P2 + 0.25 * P3 + 0.25 * P4 + 0.05 * P5)$ mayor o igual a 5, sino está suspensa.

Criterios de evaluación para los no asistentes:

Metodología / Examen: Para los no asistentes, el alumno es evaluado con un examen escrito final que cubre todo el contenido de la asignatura y mide la adquisición de las competencias de la asignatura por parte del alumno (permitiendo una muestra aleatoria para limitar el tiempo del examen a 3 horas). % Calificación: 95%. Competencias evaluadas: todas. Resultados de aprendizaje evaluados: todos.

Metodología / Prácticas: Entrega de las prácticas propuestas en las clases prácticas. % Calificación: 5%. Competencias evaluadas: todas. Resultados de aprendizaje evaluados: RA2, RA3, RA4.

Criterios de evaluación para la 2ª edición del acta:

Los estudiantes que no logren al menos un aprobado en la primera evaluación tendrán la posibilidad de intentar aprobar los apartados P2, P3 y/o P4 en la segunda edición de las actas.

Si así lo desean, los estudiantes pueden elegir ser evaluados según los criterios de evaluación para los estudiantes no asistentes.

Criterios de evaluación para las actas de fin de carrera:

Dado que los exámenes de fin de carrera se llevan a cabo al comienzo del curso, se evalúa sin asistencia con la posibilidad de evaluar el informe y las entregas de las prácticas a través del trabajo ya realizado durante el curso matriculado anterior.

El calendario de pruebas de evaluación aprobado oficialmente por la Junta del Centro ESEI se publica en el sitio web <http://www.esei.uvigo.es>. Existe coordinación entre las asignaturas del mismo curso para equilibrar el proceso de evaluación continua durante el curso.

Fuentes de información

Bibliografía Básica

J.T. Palma Méndez, M.C. Garrido Carrera, F. Sánchez Figueroa, A. Quesada Arencibia., **Programacion Concurrente**, 84-9732-184-7, OUR 681.32 /392, Thomson, 2003

D. Lea, **Programación concurrente en Java**, 8478290389, OUR 681.32 /426, Addison Wesley, 2001

G. Coulouris, J. Dollimore, T. Kindberg, **Sistemas distribuidos : conceptos y diseño**, 84-7829-049-4, OUR 681.31 /27, Addison Wesley, 2001

M.L. Liu, **Computación distribuida : fundamentos y aplicaciones**, 8478290664, OUR 681.31 /201, Pearson/Addison Wesley, 2004

M. Herlihy, N. Shavit, **The Art of Multiprocessor Programming**, 9780123973375, OUR 681.32 /538, Morgan Kaufmann, 2012

C. Breshears, **The Art of Concurrency**, 9780596521530, OUR 681.32 /559, O'Reilly, 2009

Bibliografía Complementaria

D. Schmidt, M. Stal, H. Rohnert, F. Buschmann, **Pattern-oriented Software Architecture (v.2), Pattern for Concurrent and Networked Objects**, 978-0-471-48648-0, OUR 681.321 /16, John Wiley, 2007

Varios, **Internet**, <http://www.java.com>, Oracle, 2020

Varios, **Internet**, <http://www.cppreference.com>, 2020

Recomendaciones

Otros comentarios

Según el mencionado en la memoria de la titulación se recomienda tener conocimiento de estructuras de datos y algoritmos básicos y del suyo análisis, conocimiento de entornos y herramientas de desarrollo de software, conocimiento de conceptos

básicos de la programación, conocimientos principales de la matemática, y conocer el principal funcionamiento de los sistemas operativos.

Se menciona que prácticamente todas las asignaturas optativas en un o en otro aspecto requieren el concepto de concurrencia y distribución en sistemas modernos para alcanzar sus objetivos específicos.
