



Guía docente

270602 - CPDS - Concurrencia, Paralelismo y Sistemas Distribuidos

Última modificación: 16/07/2024

Unidad responsable: Facultad de Informática de Barcelona
Unidad que imparte: 701 - DAC - Departamento de Arquitectura de Computadores.
723 - CS - Departamento de Ciencias de la Computación.

Titulación: MÁSTER UNIVERSITARIO EN INNOVACIÓN E INVESTIGACIÓN EN INFORMÁTICA (Plan 2012). (Asignatura obligatoria).

Curso: 2024 **Créditos ECTS:** 6.0 **Idiomas:** Inglés

PROFESORADO

Profesorado responsable: JORDI GUITART FERNANDEZ

Otros: Primer quadrimestre:
JORGE CASTRO RABAL - 11, 12
JORDI GUITART FERNANDEZ - 11, 12
JOSE RAMON HERRERO ZARAGOZA - 11, 12

CAPACIDADES PREVIAS

Concurrencia: conocimientos de Java a nivel de clases y objetos.
Paralelismo: conocimientos básicos sobre arquitecturas paralelas, incluyendo multiprocesadores con memoria compartida y distribuida.
Distribución: conocimientos de la estructura interna y el funcionamiento de un sistema operativo y de una red de computadores.

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:
CEE2.1. Capacidad para entender los modelos, problemas y algoritmos relacionados con los sistemas distribuidos, así como poder diseñar y evaluar algoritmos y sistemas que traten la problemática de la distribución y ofrezcan servicios distribuidos
CEE2.3. Capacidad de entender los modelos, problemas y herramientas matemáticas que permiten analizar, diseñar y evaluar redes de computadores y sistemas distribuidos.
CEE4.2. Capacidad de analizar, evaluar, diseñar y optimizar software considerando la arquitectura y de proponer nuevas técnicas de optimización.

Genéricas:
CG1. Capacidad para aplicar el método científico en el estudio y análisis de fenómenos y sistemas en cualquier ámbito de la Informática, así como en la concepción, diseño e implantación de soluciones informáticas innovadoras y originales.
CG5. Capacidad para aplicar soluciones innovadoras y realizar avances en el conocimiento que exploten los nuevos paradigmas de la Informática, particularmente en entornos distribuidos.

Transversales:
CTR3. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.
CTR6. RAZONAMIENTO: Capacidad de razonamiento crítico, lógico y matemático. Capacidad para resolver problemas dentro de su área de estudio. Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales. Capacidad de diseñar y realizar experimentos sencillos, y analizar e interpretar sus resultados. Capacidad de análisis, síntesis y evaluación.

METODOLOGÍAS DOCENTES

Durante el curso se realizarán dos tipos de actividades:

- Actividades centradas en la adquisición de conocimientos teóricos.
- Actividades centradas en la adquisición de conocimientos mediante experimentación con la implementación y evaluación empírica en el laboratorio de los mecanismos explicados a nivel teórico.

Las actividades teóricas incluyen clases expositivas participativas donde se explican los contenidos básicos del curso. Las actividades prácticas incluyen seminarios de laboratorio donde los alumnos implementan los mecanismos descritos en las clases expositivas. Los seminarios requieren de una preparación previa mediante la lectura del enunciado y la documentación de apoyo, y una elaboración posterior de las conclusiones obtenidas en un informe.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

- Entender la definición de sistema distribuido y sus posibles aplicaciones, además de los desafíos que se deben afrontar para su diseño e implementación.
- Entender la problemática del tiempo y la ordenación de eventos en un sistema distribuido y explicar e implementar los mecanismos de relojes lógicos para atacar esta problemática y los algoritmos para sincronizar relojes físicos en un sistema distribuido.
- Entender la problemática de obtener un estado global consistente en un sistema distribuido y explicar el mecanismo de snapshot distribuido para atacar esta problemática, además de definir predicados globales para la evaluación de propiedades en un sistema distribuido.
- Describir, comparar e implementar los algoritmos para la coordinación de procesos en un sistema distribuido, incluyendo la coordinación necesaria para elección de líder, comunicación en grupo multicast, y consenso.
- Comprender la problemática de la ejecución concurrente de transacciones y describir, comparar e implementar diferentes mecanismos de control de concurrencia.
- Ampliar el concepto de transacción, el protocolo de commit y los mecanismos de control de concurrencia a un sistema distribuido.
- Comprender la aplicación de la replicación en un sistema distribuido, además de la problemática que se introduce a nivel de consistencia, y describir los modelos de consistencia correspondientes y su implementación.
- Comprender los problemas que puede causar el acceso concurrente de varios agentes (threads) a los recursos. Conocer las estrategias de diseño que aseguran una correcta coordinación de los agentes y evitan las inconveniencias del acceso concurrente
- Adquirir destreza en el diseño de programas concurrentes en el paradigma de memoria compartida.
- Adquirir destreza en el diseño de programas concurrentes en el paradigma de paso de mensajes.
- Comprender y medir el paralelismo potencial disponible en una aplicación secuencial y el rendimiento logrado con su implementación paralela
- Decidir la estrategia de descomposición más adecuada para expresar el paralelismo en una aplicación
- Especificar, utilizando el paradigma de programación apropiado, una versión paralela eficiente que corresponda a una determinada descomposición de tareas / datos

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTADO

Tipo	Horas	Porcentaje
Horas actividades dirigidas	3,0	1.99
Horas aprendizaje autónomo	96,0	63.58
Horas grupo grande	26,0	17.22
Horas grupo pequeño	26,0	17.22

Dedicación total: 151 h

CONTENIDOS

Sistemas de transición y algebra de procesos

Descripción:

Procesos secuenciales y sistemas de transición. Construcciones prefix y choice. Concurrencia de procesos, concepto y construcción. Análisis de sistemas de transición. Implementaciones en Java y Erlang.



Entender el paralelismo

Descripción:

Introducción a las arquitecturas paralelas: arquitecturas de memoria compartida y distribuida y aceleradores. Gráfico de tareas: nodos y aristas. Métrica. Aceleración y eficiencia. Ley Amdahl.

Conceptos de sistemas distribuidos

Descripción:

Definición de sistema distribuido. Posibles aplicaciones de un sistema distribuido. Ejemplos de sistemas distribuidos. Desafíos para diseñar e implementar un sistema distribuido: heterogeneidad, seguridad, ausencia de visión global, concurrencia, ausencia de un único punto de control, asincronía, apertura, transparencia, tolerancia a fallos, escalabilidad.

Propiedades de seguridad y vivacidad

Descripción:

Descripción y ejemplos de propiedades de seguridad. Análisis propiedades de seguridad sobre sistemas de transición. Descripción de las propiedades de vivacidad y especialmente de las propiedades de progreso. Análisis de propiedades de progreso en sistemas de transición.

Concurrencia, exclusion mutua y condiciones de sincronización. Deadlock.

Descripción:

El problema de la interferencia destructiva. Locks y exclusión mutua. Semáforos, monitores y condiciones de sincronización. El problema del deadlock y su análisis sobre sistemas de transición. Estrategias de diseño para impedir deadlocks.

Paso de mensajes. Arquitecturas

Descripción:

Descripción del paradigma de paso de mensajes. La arquitectura cliente/servidor. Introducción a otras arquitecturas: Filter pipeline, supervisor/worker, announcer/listener. Paso de mensajes en Erlang. Diseños de arquitecturas en Erlang

Predicción y análisis del rendimiento

Descripción:

Uso de modelos y herramientas para entender el paralelismo y el rendimiento (Tareador, Extrae, Paraver y Dimemas).

Programación de memoria distribuida usando MPI

Descripción:

Visión general de la arquitectura del clúster. Creación, identificación y terminación de procesos. Operaciones punto a punto vs. operaciones colectivas. Comunicaciones síncronas vs asíncronas.



Programación de dispositivos GPU para la aceleración computacional usando CUDA

Descripción:

Descripción de la arquitectura de la GPU. Decomposiciones adecuadas para la aceleración de la GPU. Principios de programación de CUDA.

Algoritmos distribuidos: Tiempo, estado global, coordinación y consenso

Descripción:

Tiempo y la ordenación de eventos en un sistema distribuido. Relojes lógicos: relación happened-before, relojes lógicos de Lamport (escalares, vectoriales). Algoritmos para sincronizar relojes físicos en un sistema distribuido: Cristian (NTP), Berkeley. Estado global consistente en un sistema distribuido. Mecanismo de snapshot distribuido de Chandy y Lamport. Predicados globales para la evaluación de propiedades en un sistema distribuido: propiedades de los predicados (estabilidad), ocurrencia de los predicados (possibly, definitely). Coordinación de procesos en un sistema distribuido para la elección de líder: Bully, Ring. Coordinación de procesos en un sistema distribuido para la comunicación en grupo multicast: multicast fiable básico, multicast fiable escalable, multicast ordenado (FIFO, causal, total), atomic multicast. Coordinación de procesos en un sistema distribuido para garantizar el consenso: problema de los dos ejércitos, algoritmo de Dolev & Strong, problema de los generales Bizantinos, Paxos

Datos compartidos distribuidos: Transacciones, consistencia y replicación

Descripción:

Ejecución concurrente de transacciones: lost update, inconsistent retrievals, equivalencia serie, recuperación de aborts (dirty read, premature write). Mecanismos de control de concurrencia: two-phase locking (incluyendo detección y tratamiento de deadlock), optimistic concurrency control, timestamp ordering. Transacción distribuida. Protocolos de commit distribuido: one-phase y two-phase. Mecanismos de control de concurrencia en un sistema distribuido: two-phase locking (incluyendo detección y tratamiento de deadlock distribuido), optimistic concurrency control, timestamp ordering. Replicación y consistencia en un sistema distribuido. Modelos de consistencia fuerte centrados en los datos: estricta, linearizability, secuencial. Modelos de consistencia relajada centrados en los datos: uso de variables de sincronización. Modelos de consistencia centrados en el cliente: eventual, monotonic-read, monotonic-write, read-your-writes, writes-follow-reads. Implementaciones de modelos de consistencia: protocolos basados en primario (remote-write, local-write) y protocolos de escritura replicada (replicación activa, protocolos basados en quórum)

ACTIVIDADES

Conceptos fundamentales de concurrencia, paralelismo y distribución

Descripción:

Preparación de la clase con la ayuda del material de apoyo. Comprensión y asimilación de los contenidos del tema para su aplicación posterior

Objetivos específicos:

1, 8, 11

Competencias relacionadas:

CEE2.1. Capacidad para entender los modelos, problemas y algoritmos relacionados con los sistemas distribuidos, así como poder diseñar y evaluar algoritmos y sistemas que traten la problemática de la distribución y ofrezcan servicios distribuidos

CEE4.2. Capacidad de analizar, evaluar, diseñar y optimizar software considerando la arquitectura y de proponer nuevas técnicas de optimización.

CG5. Capacidad para aplicar soluciones innovadoras y realizar avances en el conocimiento que exploten los nuevos paradigmas de la Informática, particularmente en entornos distribuidos.

CG1. Capacidad para aplicar el método científico en el estudio y análisis de fenómenos y sistemas en cualquier ámbito de la Informática, así como en la concepción, diseño e implantación de soluciones informáticas innovadoras y originales.

Dedicación: 18h

Aprendizaje autónomo: 12h

Grupo grande/Teoría: 6h

Primer modulo elegido: Concurrencia o Paralelismo o Distribución

Descripción:

Preparación de la clase con la ayuda del material de apoyo. Comprensión y asimilación de los contenidos asociados a los temas del modulo correspondiente para su aplicación posterior en las sesiones de laboratorio.

Dedicación: 50h

Aprendizaje autónomo: 30h

Grupo grande/Teoría: 10h

Grupo pequeño/Laboratorio: 10h

Segundo modulo elegido: Concurrencia o Paralelismo o Distribución

Descripción:

Preparación de la clase con la ayuda del material de apoyo. Comprensión y asimilación de los contenidos asociados a los temas del modulo correspondiente para su aplicación posterior en las sesiones de laboratorio.

Dedicación: 50h

Aprendizaje autónomo: 30h

Grupo grande/Teoría: 10h

Grupo pequeño/Laboratorio: 10h

Módulo I: Concurrencia

Descripción:

Preparación de la clase con la ayuda del material de apoyo. Comprensión y asimilación de los contenidos asociados a los temas del Módulo I (concurrencia) para su aplicación posterior en las sesiones de laboratorio.

Objetivos específicos:

8, 9, 10

Competencias relacionadas:

CEE2.1. Capacidad para entender los modelos, problemas y algoritmos relacionados con los sistemas distribuidos, así como poder diseñar y evaluar algoritmos y sistemas que traten la problemática de la distribución y ofrezcan servicios distribuidos

CEE4.2. Capacidad de analizar, evaluar, diseñar y optimizar software considerando la arquitectura y de proponer nuevas técnicas de optimización.

CEE2.3. Capacidad de entender los modelos, problemas y herramientas matemáticas que permiten analizar, diseñar y evaluar redes de computadores y sistemas distribuidos.

CG1. Capacidad para aplicar el método científico en el estudio y análisis de fenómenos y sistemas en cualquier ámbito de la Informática, así como en la concepción, diseño e implantación de soluciones informáticas innovadoras y originales.

CTR3. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Módulo II: Paralelismo

Descripción:

Preparación de la clase con la ayuda del material de apoyo. Comprensión y asimilación de los contenidos asociados a los temas del Módulo II (paralelismo) para su aplicación posterior en las sesiones de laboratorio.

Objetivos específicos:

11, 12, 13

Competencias relacionadas:

CEE4.2. Capacidad de analizar, evaluar, diseñar y optimizar software considerando la arquitectura y de proponer nuevas técnicas de optimización.

CG5. Capacidad para aplicar soluciones innovadoras y realizar avances en el conocimiento que exploten los nuevos paradigmas de la Informática, particularmente en entornos distribuidos.

CG1. Capacidad para aplicar el método científico en el estudio y análisis de fenómenos y sistemas en cualquier ámbito de la Informática, así como en la concepción, diseño e implantación de soluciones informáticas innovadoras y originales.

CTR6. RAZONAMIENTO: Capacidad de razonamiento crítico, lógico y matemático. Capacidad para resolver problemas dentro de su área de estudio. Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales. Capacidad de diseñar y realizar experimentos sencillos, y analizar e interpretar sus resultados. Capacidad de análisis, síntesis y evaluación.

CTR3. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Módulo III: Distribución

Descripción:

Preparación de la clase con la ayuda del material de apoyo. Comprensión y asimilación de los contenidos asociados a los temas del Módulo III (distribución) para su aplicación posterior en las sesiones de laboratorio.

Objetivos específicos:

1, 2, 3, 4, 5, 6, 7

Competencias relacionadas:

CEE2.1. Capacidad para entender los modelos, problemas y algoritmos relacionados con los sistemas distribuidos, así como poder diseñar y evaluar algoritmos y sistemas que traten la problemática de la distribución y ofrezcan servicios distribuidos

CEE2.3. Capacidad de entender los modelos, problemas y herramientas matemáticas que permiten analizar, diseñar y evaluar redes de computadores y sistemas distribuidos.

CG5. Capacidad para aplicar soluciones innovadoras y realizar avances en el conocimiento que exploten los nuevos paradigmas de la Informática, particularmente en entornos distribuidos.

CTR6. RAZONAMIENTO: Capacidad de razonamiento crítico, lógico y matemático. Capacidad para resolver problemas dentro de su área de estudio. Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales. Capacidad de diseñar y realizar experimentos sencillos, y analizar e interpretar sus resultados. Capacidad de análisis, síntesis y evaluación.

CTR3. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Primer modulo elegido: Preparación examen

Descripción:

Repaso general y preparación examen final

Objetivos específicos:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

Competencias relacionadas:

CEE2.1. Capacidad para entender los modelos, problemas y algoritmos relacionados con los sistemas distribuidos, así como poder diseñar y evaluar algoritmos y sistemas que traten la problemática de la distribución y ofrezcan servicios distribuidos

CEE4.2. Capacidad de analizar, evaluar, diseñar y optimizar software considerando la arquitectura y de proponer nuevas técnicas de optimización.

CEE2.3. Capacidad de entender los modelos, problemas y herramientas matemáticas que permiten analizar, diseñar y evaluar redes de computadores y sistemas distribuidos.

CG5. Capacidad para aplicar soluciones innovadoras y realizar avances en el conocimiento que exploten los nuevos paradigmas de la Informática, particularmente en entornos distribuidos.

CG1. Capacidad para aplicar el método científico en el estudio y análisis de fenómenos y sistemas en cualquier ámbito de la Informática, así como en la concepción, diseño e implantación de soluciones informáticas innovadoras y originales.

CTR6. RAZONAMIENTO: Capacidad de razonamiento crítico, lógico y matemático. Capacidad para resolver problemas dentro de su área de estudio. Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales. Capacidad de diseñar y realizar experimentos sencillos, y analizar e interpretar sus resultados. Capacidad de análisis, síntesis y evaluación.

CTR3. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 6h

Aprendizaje autónomo: 4h

Actividades dirigidas: 2h



Segundo modulo elegido: Preparación examen

Descripción:

Repaso general y preparación examen final

Objetivos específicos:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

Competencias relacionadas:

CEE2.1. Capacidad para entender los modelos, problemas y algoritmos relacionados con los sistemas distribuidos, así como poder diseñar y evaluar algoritmos y sistemas que traten la problemática de la distribución y ofrezcan servicios distribuidos

CEE4.2. Capacidad de analizar, evaluar, diseñar y optimizar software considerando la arquitectura y de proponer nuevas técnicas de optimización.

CEE2.3. Capacidad de entender los modelos, problemas y herramientas matemáticas que permiten analizar, diseñar y evaluar redes de computadores y sistemas distribuidos.

CG5. Capacidad para aplicar soluciones innovadoras y realizar avances en el conocimiento que exploten los nuevos paradigmas de la Informática, particularmente en entornos distribuidos.

CG1. Capacidad para aplicar el método científico en el estudio y análisis de fenómenos y sistemas en cualquier ámbito de la Informática, así como en la concepción, diseño e implantación de soluciones informáticas innovadoras y originales.

CTR6. RAZONAMIENTO: Capacidad de razonamiento crítico, lógico y matemático. Capacidad para resolver problemas dentro de su área de estudio. Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales. Capacidad de diseñar y realizar experimentos sencillos, y analizar e interpretar sus resultados. Capacidad de análisis, síntesis y evaluación.

CTR3. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 6h

Aprendizaje autónomo: 4h

Actividades dirigidas: 2h

Primer modulo elegido: Examen final

Descripción:

Asimilación de los conceptos del curso y realización del examen

Objetivos específicos:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

Competencias relacionadas:

CEE2.1. Capacidad para entender los modelos, problemas y algoritmos relacionados con los sistemas distribuidos, así como poder diseñar y evaluar algoritmos y sistemas que traten la problemática de la distribución y ofrezcan servicios distribuidos

CEE4.2. Capacidad de analizar, evaluar, diseñar y optimizar software considerando la arquitectura y de proponer nuevas técnicas de optimización.

CEE2.3. Capacidad de entender los modelos, problemas y herramientas matemáticas que permiten analizar, diseñar y evaluar redes de computadores y sistemas distribuidos.

CG5. Capacidad para aplicar soluciones innovadoras y realizar avances en el conocimiento que exploten los nuevos paradigmas de la Informática, particularmente en entornos distribuidos.

CG1. Capacidad para aplicar el método científico en el estudio y análisis de fenómenos y sistemas en cualquier ámbito de la Informática, así como en la concepción, diseño e implantación de soluciones informáticas innovadoras y originales.

CTR6. RAZONAMIENTO: Capacidad de razonamiento crítico, lógico y matemático. Capacidad para resolver problemas dentro de su área de estudio. Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales. Capacidad de diseñar y realizar experimentos sencillos, y analizar e interpretar sus resultados. Capacidad de análisis, síntesis y evaluación.

CTR3. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 10h

Aprendizaje autónomo: 8h

Actividades dirigidas: 2h

Segundo modulo elegido: Examen final

Descripción:

Asimilación de los conceptos del curso y realización del examen

Objetivos específicos:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

Competencias relacionadas:

CEE2.1. Capacidad para entender los modelos, problemas y algoritmos relacionados con los sistemas distribuidos, así como poder diseñar y evaluar algoritmos y sistemas que traten la problemática de la distribución y ofrezcan servicios distribuidos

CEE4.2. Capacidad de analizar, evaluar, diseñar y optimizar software considerando la arquitectura y de proponer nuevas técnicas de optimización.

CEE2.3. Capacidad de entender los modelos, problemas y herramientas matemáticas que permiten analizar, diseñar y evaluar redes de computadores y sistemas distribuidos.

CG5. Capacidad para aplicar soluciones innovadoras y realizar avances en el conocimiento que exploten los nuevos paradigmas de la Informática, particularmente en entornos distribuidos.

CG1. Capacidad para aplicar el método científico en el estudio y análisis de fenómenos y sistemas en cualquier ámbito de la Informática, así como en la concepción, diseño e implantación de soluciones informáticas innovadoras y originales.

CTR6. RAZONAMIENTO: Capacidad de razonamiento crítico, lógico y matemático. Capacidad para resolver problemas dentro de su área de estudio. Capacidad de abstracción: capacidad de crear y utilizar modelos que reflejen situaciones reales. Capacidad de diseñar y realizar experimentos sencillos, y analizar e interpretar sus resultados. Capacidad de análisis, síntesis y evaluación.

CTR3. TRABAJO EN EQUIPO: Ser capaz de trabajar como miembro de un equipo, ya sea como un miembro más, o realizando tareas de dirección con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 10h

Aprendizaje autónomo: 8h

Actividades dirigidas: 2h

SISTEMA DE CALIFICACIÓN

Para cada modulo, habrá un examen (EF) y una nota de laboratorio (L). El examen será de problemas sobre la teoría explicada. La nota de laboratorio reflejará la calidad del trabajo realizado en las prácticas de laboratorio. La nota final del modulo será $NF=0.6*EF+0.4*L$.

La nota final se calculará como la media aritmética de la nota de los dos modulos cursados por el estudiante: $NF=0.5*M1+0.5*M2$.

BIBLIOGRAFÍA

Básica:

- Tanenbaum, A.S.; Steen, M. van. Distributed systems: principles and paradigms. 2nd ed. Pearson Prentice Hall, 2007. ISBN 0136135536.
- Coulouris, G.F.; Dollimore, J.; Kindberg, T.; Blair, G. Distributed systems: concepts and design. 5th ed., int. ed. Addison-Wesley/Pearson Education, 2012. ISBN 0273760599.
- Magee, J.; Kramer, J. Concurrency: state models & Java programs. 2nd ed. John Wiley & Sons, 2006. ISBN 0470093552.
- Armstrong, J. Programming Erlang: software for a concurrent world. 2nd ed. Raleigh, N.C: Pragmatic Bookshelf, 2013. ISBN 9781937785536.
- Goetz, B.; Peierls, T.; Bloch, J.; Bowbeer, J.; Holmes, D.; Lea, D. Java concurrency in practice. Addison-Wesley, 2006. ISBN 0321349601.
- Grama, A.; Karypis, G.; Kumar, V.; Gupta, A. Introduction to parallel computing. 2nd ed. Pearson Education, 2003. ISBN 0201648652.

Complementaria:

- Cesarini, F.; Thompson, S. Erlang programming. O'Reilly, 2009. ISBN 9780596518189.
- Herlihy, M. [i 3 més]. The art of multiprocessor programming. Second edition. Cambridge, MA: Morgan Kaufmann Publishers is an

imprint of Elsevier, 2021. ISBN 9780124159501.