



UNIVERSIDAD
POLITÉCNICA
DE MADRID

PROCESO DE
COORDINACIÓN DE LAS
ENSEÑANZAS PR/CL/001



E.T.S. de Ingeniería de
Sistemas Informáticos

ANX-PR/CL/001-01

GUÍA DE APRENDIZAJE

ASIGNATURA

613000112 - Sistemas Distribuidos Tolerantes A Fallos

PLAN DE ESTUDIOS

61AG - Master Universitario En Software De Sistemas Distribuidos Y Empotrados

CURSO ACADÉMICO Y SEMESTRE

2024/25 - Segundo semestre

Índice

Guía de Aprendizaje

1. Datos descriptivos.....	1
2. Profesorado.....	1
3. Conocimientos previos recomendados.....	2
4. Competencias y resultados de aprendizaje.....	3
5. Descripción de la asignatura y temario.....	4
6. Cronograma.....	6
7. Actividades y criterios de evaluación.....	8
8. Recursos didácticos.....	10
9. Otra información.....	11

1. Datos descriptivos

1.1. Datos de la asignatura

Nombre de la asignatura	613000112 - Sistemas Distribuidos Tolerantes a Fallos
No de créditos	3 ECTS
Carácter	Obligatoria
Curso	Primer curso
Semestre	Segundo semestre
Período de impartición	Febrero-Junio
Idioma de impartición	Castellano
Titulación	61AG - Master Universitario en Software de Sistemas Distribuidos y Empotrados
Centro responsable de la titulación	61 - Escuela Tecnica Superior De Ingenieria De Sistemas Informaticos
Curso académico	2024-25

2. Profesorado

2.1. Profesorado implicado en la docencia

Nombre	Despacho	Correo electrónico	Horario de tutorías *
Gustavo Adolfo Hernandez Peñalozza (Coordinador/a)	4408	gustavo.hernandez.penalozza@upm.es	Sin horario. Las tutorías se conciertan con los alumnos para facilitar su asistencia.

Silvia Alba Uribe Mayoral	4218	silviaalba.uribe@upm.es	Sin horario. Las tutorías se conciertan con los alumnos para facilitar su asistencia
Juan Antonio Rodrigo Ferran	4201	juanantonio.rodrido@upm.es	Sin horario. Las tutorías se conciertan con los alumnos para facilitar su asistencia

* Las horas de tutoría son orientativas y pueden sufrir modificaciones. Se deberá confirmar los horarios de tutorías con el profesorado.

3. Conocimientos previos recomendados

3.1. Asignaturas previas que se recomienda haber cursado

El plan de estudios Master Universitario en Software de Sistemas Distribuidos y Empotrados no tiene definidas asignaturas previas recomendadas para esta asignatura.

3.2. Otros conocimientos previos recomendados para cursar la asignatura

- Sistemas Distribuidos. Programación. Sistemas Operativos. Redes.

4. Competencias y resultados de aprendizaje

4.1. Competencias

CE02 - Capacidad para la dirección de proyectos de desarrollo e innovación del ámbito de los Sistemas Distribuidos y Empotrados, en empresas y centros tecnológicos, con garantía de la seguridad para las personas y bienes, y la calidad final de los productos.

CE04 - Capacidad de comprender y saber aplicar el funcionamiento y organización de Internet, las tecnologías y protocolos de redes de nueva generación, los modelos de componentes, software intermediario y los servicios.

CE05 - Capacidad para diseñar, desarrollar, gestionar y evaluar mecanismos de garantía de seguridad en el tratamiento y acceso a la información en un sistema de procesamiento local o distribuido.

CE06 - Capacidad para diseñar y evaluar aplicaciones y sistemas basados en computación distribuida y para implantar sistemas operativos y servidores.

4.2. Resultados del aprendizaje

RA69 - RA2 - Construye servicios y aplicaciones distribuidas coherentes y disponibles ante fallos de datos y procesos.

RA70 - RA1 - Identifica y comprende los modelos y arquitecturas actuales de los sistemas distribuidos tolerantes a fallos.

RA71 - RA3 - Diseña algoritmos de sincronización y coordinación distribuidos tolerantes a fallos.

RA72 - RA4 - Especifica, diseña e implementa en un proyecto una aplicación o servicio web distribuido y tolerante a fallos.

RA68 - RA80 - Sistemáticamente revisa su trabajo y la forma de hacerlo, reduce errores e introduce mejoras

5. Descripción de la asignatura y temario

5.1. Descripción de la asignatura

Cada vez más nuestra sociedad depende de los computadores. En algunos casos esta dependencia es crítica ya que del buen funcionamiento del sistema de computación dependen vidas humanas o supone una gran cantidad de pérdidas económicas o de negocio. Esta dependencia se ve comprometida si fallan los elementos hardware del sistema. Ejemplos típicos de este tipo de dependencia crítica son: los sistemas de control de tráfico aéreo y de trenes, los sistemas de control de centrales nucleares, sistemas hospitalarios, las grandes web de sistemas de reservas, bancos, contratación en bolsa, etc. Este grado de dependencia hace que tengan que existir técnicas que garanticen un alto grado de fiabilidad y disponibilidad en los sistemas de computación. El problema se puede abordar desde el hardware y desde el software. En ambos casos se utiliza redundancia para conseguir tolerar fallos de hardware. Las técnicas hardware son en general muy caras debido al reducido nivel de producción que obliga a repercutir los costos de I+D en menos cantidad de productos. En cambio las técnicas software permiten una mayor versatilidad y adaptabilidad a diferentes hardware.

Este curso aborda el problema de cómo construir sistemas de computación que garanticen unos grados muy altos de fiabilidad y disponibilidad a las aplicaciones críticas usando técnicas software para tolerar fallos hardware. Para conseguir estas garantías estudiaremos como enmascarar fallos hardware (tolerancia a fallos). Veremos como utilizar la replicación de datos y de procesos. Usaremos como arquitectura de computación el sistema distribuido que nos garantiza redundancia intrínseca con probabilidad de fallos independiente en el funcionamiento de sus componentes. Para conseguir este enmascaramiento es necesario conocer los diferentes tipos de coherencia en el acceso a réplicas. Estudiaremos por tanto las diferentes semánticas de coherencia tanto en procesos como en datos. Estudiaremos en detalle dos de estas semánticas, la atómica (Raft, Zookeeper) y la eventual (Ethereum).

5.2. Temario de la asignatura

1. Coherencia en replicación de procesos y datos.
 - 1.1. Modelo
 - 1.2. Coherencia atómica
 - 1.3. Coherencia secuencia
 - 1.4. Coherencia causal
 - 1.5. Coherencia eventual
2. Teorema de imposibilidad CAP
3. Replicación con coherencia atómica. Raft.
 - 3.1. Introducción
 - 3.2. Modelo
 - 3.3. Elección de líder
 - 3.4. Replicación de log
 - 3.5. Garantizando la propiedad de safety
 - 3.6. Coherencia atómica
4. Replicación con coherencia eventual. Blockchain Ethereum.
 - 4.1. Red peer-to-peer
 - 4.2. Proof of work. Consenso tarde o temprano.
 - 4.3. Replicación con coherencia eventual.
5. Laboratorio. Coherencia atómica. Zookeeper y Raft.
 - 5.1. Introducción
 - 5.2. Interfaz del servicio mediante comandos
 - 5.3. Programación del servicio con el API de Java

6. Cronograma

6.1. Cronograma de la asignatura *

Sem	Actividad tipo 1	Actividad tipo 2	Tele-enseñanza	Actividades de evaluación
1				
2				
3				
4	<p>Tema 1. Consistencia en la replicación. Duración: 01:15 LM: Actividad del tipo Lección Magistral</p> <p>Tema 2. Teorema de imposibilidad CAP. Duración: 00:15 LM: Actividad del tipo Lección Magistral</p> <p>Tema 3. Introducción a Raft. Elección de líder. Duración: 01:00 LM: Actividad del tipo Lección Magistral</p>	<p>Tolerancia a fallos con Zookeeper y Kafka. Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio</p>	<p>Tutorías on-line Duración: 05:00 OT: Otras actividades formativas / Evaluación</p>	
5	<p>Tema 3. Raft. Replicación de logs. Consistencia. Tolerancia a fallos. Duración: 01:15 LM: Actividad del tipo Lección Magistral</p> <p>Tema 4. Ethereum. Redes peer-to-peer. Consenso eventual. Duración: 01:15 LM: Actividad del tipo Lección Magistral</p>	<p>Tolerancia a fallos con Zookeeper y Kafka. Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio</p>	<p>Tutorías on-line Duración: 05:00 OT: Otras actividades formativas / Evaluación</p>	
6	<p>Tema 4. Ethereum. Coherencia eventual. Duración: 01:30 LM: Actividad del tipo Lección Magistral</p>	<p>Tolerancia a fallos con Zookeeper y Kafka. Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio</p>	<p>Tutorías on-line Duración: 05:00 OT: Otras actividades formativas / Evaluación</p>	<p>EX-TE-Evaluación progresiva. Teoría. Temas 1 a 4. (RA1, RA2, RA3) EX: Técnica del tipo Examen Escrito Evaluación Progresiva Presencial Duración: 01:00</p> <p>EX-PR-Evaluación progresiva. Practicas. (RA4, RA80) EP: Técnica del tipo Examen de Prácticas Evaluación Progresiva Presencial Duración: 00:30</p>
7				
8				
9				
10				

11				
12				
13				
14				
15				
16				
17				<p>Evaluación global. Teoría. Temas 1 a 4. (RA1, RA2, RA3) EX: Técnica del tipo Examen Escrito Evaluación Global Presencial Duración: 01:00</p> <p>Evaluación global. Prácticas. (RA4, RA80) EP: Técnica del tipo Examen de Prácticas Evaluación Global Presencial Duración: 00:30</p>

Para el cálculo de los valores totales, se estima que por cada crédito ECTS el alumno dedicará dependiendo del plan de estudios, entre 26 y 27 horas de trabajo presencial y no presencial.

7. Actividades y criterios de evaluación

7.1. Actividades de evaluación de la asignatura

7.1.1. Evaluación (progresiva)

Sem.	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
6	EX-TE-Evaluación progresiva. Teoría. Temas 1 a 4. (RA1, RA2, RA3)	EX: Técnica del tipo Examen Escrito	Presencial	01:00	50%	0 / 10	CE02 CE04 CE05 CE06
6	EX-PR-Evaluación progresiva. Practicas. (RA4, RA80)	EP: Técnica del tipo Examen de Prácticas	Presencial	00:30	50%	0 / 10	CE02 CE04 CE05 CE06

7.1.2. Prueba evaluación global

Sem	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
17	Evaluación global. Teoría. Temas 1 a 4. (RA1, RA2, RA3)	EX: Técnica del tipo Examen Escrito	Presencial	01:00	50%	0 / 10	CE02 CE04 CE05 CE06
17	Evaluación global. Prácticas. (RA4, RA80)	EP: Técnica del tipo Examen de Prácticas	Presencial	00:30	50%	0 / 10	CE02 CE04 CE05 CE06

7.1.3. Evaluación convocatoria extraordinaria

Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
Evaluación global de teoría (RA1, RA2, RA3)	EX: Técnica del tipo Examen Escrito	Presencial	02:00	100%	5 / 10	CE02 CE04 CE05 CE06

7.2. Criterios de evaluación

EVALUACION EN LA CONVOCATORIA ORDINARIA

a) Evaluación progresiva de la asignatura

La evaluación progresiva de la asignatura se compone de los siguientes bloques:

Bloque I. Teoría. 1 prueba escrita individual (50%)

Bloque II. Prácticas. 1 prueba escrita individual (50%)

La nota final de la asignatura por evaluación progresiva se calcula con el siguiente algoritmo:

notaTeoria := exTeoria * 0.5

notaPracticas := exPrac * 0.5

notaTotal := **notaTeoría** + **notaPracticas**

If **notaTotal** >= 5:

notaFinal := **notaTotal**

else

notaFinal := "suspenseo"

b) Evaluación mediante prueba global

La evaluación por prueba global consta de 2 bloques evaluables: teoría y prácticas. Los alumnos cuya nota final mediante evaluación progresiva sea suspenseo (**notaFinal** = suspenseo) pero que hayan obtenido una nota igual o mayor que 4 sobre 10 en la nota de prácticas (**notaPracticas** >= 4) o en la nota de teoría (**notaTeoria** >= 4) no tendrán que examinarse de dicho bloque de nuevo.

La evaluación por prueba global se compone de los siguientes bloques:

Bloque I. Teoría. 1 prueba escrita individual (50%)

Bloque II. Prácticas. 1 prueba escrita individual (50%)

La nota final de la asignatura por prueba global se calcula con el siguiente algoritmo:

if **notaTeoria** = 5.

8. Recursos didácticos

8.1. Recursos didácticos de la asignatura

Nombre	Tipo	Observaciones
Artículo sobre Raft.	Bibliografía	D. Ongaro, J. Ousterhout. In Search of an Understandable Consensus Algorithm. USENIX Annual Technical Conference (ATC), Philadelphia, PA, 2014
Libro de Zookeeper para las prácticas.	Bibliografía	ZooKeeper: Distributed Process Coordination. Flavio Junqueira, Benjamin Reed. O'Reilly, 2013.
Moodle de la asignatura	Recursos web	moodle upm
Artículo sobre consenso en Blockchain	Bibliografía	From blockchain consensus back to Byzantine consensus. Vincent Gramoli. Future Generation Computer Systems, V107, June 2020.
Artículo sobre Ethereum	Bibliografía	Mastering Ethereum. Andreas Antonopoulos, Gavin Wood. O'Reilly 2018.
Artículo sobre coherencia de réplicas.	Bibliografía	A parametrized algorithm that implements sequential, causal, and cache memory consistencies. Ernesto Jiménez, Antonio Fernández, Vicent Cholvi. The Journal of Systems and Software 81 (2008).

Artículo sobre coherencia de réplicas en el sistema de blockchain Ethereum.	Bibliografía	On Replica Eventual Consistency of Ethereum Blockchain System. Sergio Arévalo, Isabel Muñoz. XXVII Jornadas de Concurrencia y Sistemas Distribuidos. 2021. Barcelona.
-----------------------------------------------------------------------------	--------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------

9. Otra información

9.1. Otra información sobre la asignatura

1. ACTIVIDADES SEMIPRESENCIALES

Este máster, al ser semipresencial, ofrece algunas tutorías on-line, que aparecen reflejadas en "Otra actividad presencial" en el cronograma de actividades.