



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

613000110 - Sistemas Distribuidos

PLAN DE ESTUDIOS

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

CURSO ACADÉMICO Y SEMESTRE

2024/25 - Primer semestre

Índice

Guía de Aprendizaje

1. Datos descriptivos.....	1
2. Profesorado.....	1
3. Conocimientos previos recomendados.....	2
4. Competencias y resultados de aprendizaje.....	2
5. Descripción de la asignatura y temario.....	4
6. Cronograma.....	7
7. Actividades y criterios de evaluación.....	9
8. Recursos didácticos.....	15
9. Otra información.....	16

1. Datos descriptivos

1.1. Datos de la asignatura

Nombre de la asignatura	613000110 - Sistemas Distribuidos
No de créditos	6 ECTS
Carácter	Obligatoria
Curso	Primer curso
Semestre	Primer semestre
Período de impartición	Septiembre-Enero
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 *
Silvia Alba Uribe Mayoral	D-4218	silviaalba.uribe@upm.es	Sin horario. Se acuerdan con los alumnos según sus preferencias.
Gustavo Adolfo Hernandez Peñaloza (Coordinador/a)	4408	gustavo.hernandez.penaloz@upm.es	Sin horario. Se acuerdan con los estudiantes según sus preferencias

Juan Antonio Rodrigo Ferran	4102	juanantonio.rodrido@upm.es	Sin horario. Se acuerdan con los alumnos según sus preferencias
-----------------------------	------	----------------------------	--

* 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

- Redes de Computadores
- Programación orientada a objetos
- Sistemas Operativos.

4. Competencias y resultados de aprendizaje

4.1. Competencias

CE03 - Capacidad para modelar, diseñar, definir la arquitectura, implantar, gestionar, operar, administrar y mantener aplicaciones, redes, sistemas y servicios en el ámbito de los Sistemas Distribuidos y Empotrados.

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.

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

CE07 - Capacidad de diseñar y desarrollar aplicaciones y servicios informáticos en sistemas móviles, de tiempo real, empotrados y ubicuos.

CG10 - Resolución de problemas.

CG11 - Razonamiento crítico.

CG12 - Aprendizaje autónomo, adaptación a nuevas situaciones y motivación por el desarrollo profesional permanente.

4.2. Resultados del aprendizaje

RA61 - RA39 - Construye sistemas de ficheros distribuidos

RA64 - RA43 - Construye soluciones distribuidas flexibles y escalables de calidad

RA66 - RA86- Trabajando en equipo, propone y construye soluciones a problemas en diferentes campos desde una perspectiva global

RA62 - RA38 - Construye middlewares basados tanto en comunicación directa (RPC, ...) como en comunicación indirecta (multienvío).

RA63 - RA40 - Diseña algoritmos de sincronización y coordinación distribuidos.

RA65 - RA36 - Identifica y comprende los modelos y arquitecturas actuales de los sistemas distribuidos

RA67 - RA89 - Integrar diversas teorías o modelos (de una disciplina) haciendo una síntesis personal y creativa adaptada a las propias necesidades profesionales.

RA18 - Identifica, comprende los algoritmos y protocolos utilizados en los sistemas distribuidos y en el nivel de aplicación de Internet

5. Descripción de la asignatura y temario

5.1. Descripción de la asignatura

El software que coordina un grupo de computadores conectados entre sí con redes de comunicación para obtener un objetivo concreto se denomina Sistema Distribuido. La mayoría de las aplicaciones y sistemas software que se usan actualmente son por tanto distribuidos.

En esta asignatura estudiaremos paradigmas de procesamiento distribuido muy actuales y comunes en Internet de las cosas (IoT), Cloud Computing, Blockchain y Big Data. Presentaremos primero al alumno los modelos y las arquitecturas de los sistemas distribuidos modernos. Después estudiaremos las diferentes formas en que los procesos se comunican en estos sistemas, tanto de forma directa como de forma indirecta (RPCs, RMIs, Publisher/Subscriber, Multicast). A continuación, se hablará de "Stream Processing" en los sistemas distribuidos, como forma de procesamiento continuo de datos en "tiempo-real" a medida que estos datos se producen o se reciben. Después se introducen los fundamentos de la arquitectura Blockchain Ethereum para un entorno no tolerante a fallos (un solo servidor). Finalmente, se darán los conceptos básicos para entender el funcionamiento e implementación de un sistema distribuido para el entrenamiento de Inteligencia artificial conocido como Federated Learning. En el laboratorio se construirán aplicaciones distribuidas compuestas por elementos software y Hardware simulados con NodeRed, que usan el protocolo MQTT y Apache Kafka. El protocolo MQTT es el protocolo estándar usado en Internet of Things en áreas tales como automoción, telecomunicaciones, industria, gaseoductos y oleoductos, etc.. Es un protocolo que implementa un servicio de tipo "publish/subscribe" para ser usado por una red de dispositivos donde es necesario salvaguardar el gasto energético y el ancho de banda usado y tolerar fallos de falta de conexión o de batería. Por otra parte, se explicará y usará la herramienta Kafka, como herramienta de event stream processing, que corriendo en un clúster de servidores (en cloud pública o privada) es capaz de soportar el flujo de trillones de eventos diarios para su procesamiento en tiempo real o batch. Esta herramienta se usa en plataformas con LinkedIn o Netfiix.

5.2. Temario de la asignatura

1. Introduction. System Models
 - 1.1. Definition
 - 1.2. Distributed Abstractions
 - 1.3. Examples
 - 1.4. Models
2. Indirect Communication. Publisher / subscriber.
 - 2.1. Definition of Publish/Subscribe Systems
 - 2.2. Examples
 - 2.3. Properties
 - 2.4. Programming Model
 - 2.5. Implementation
3. Stream processing
 - 3.1. Case Study Definition
 - 3.2. Stream Aggregation
 - 3.3. Event Sourcing
 - 3.4. Separating DB Reads and Writes
 - 3.5. Immutable Facts and The Source of Truth
 - 3.6. Using Append.Only Streams of Immutable Events
4. Blockchain Ethereum
 - 4.1. Introduction
 - 4.2. Cryptography
 - 4.3. Clients. Transactions
 - 4.4. Servers. Miners. Transaction validation, block execution
 - 4.5. Smart contracts
 - 4.6. World state
5. Federated Systems
 - 5.1. Introduction

- 5.2. Architectures
- 5.3. Implementation and tools
- 5.4. Federated Learning
- 6. Lab 0. Linux Systems & Object Oriented Programming
 - 6.1. Distributions and main shell commands.
 - 6.2. Classes and Objects
 - 6.3. Input /Output
 - 6.4. Processes and Threads
 - 6.5. Example of Thread program
 - 6.6. TCP Sockets
- 7. Lab 2. NodeRED Tool
 - 7.1. Programming Tool Description
 - 7.2. Examples
- 8. Lab 3. MQTT
 - 8.1. Protocol Description
 - 8.2. Quality of Service Semantics
 - 8.3. Example with Mobile Sensors, NodeRED and MQTT
- 9. Lab 4. Stream Processing with Kafka
 - 9.1. Definition
 - 9.2. Installation
 - 9.3. Kafka Command Interface
 - 9.4. Programming with Java API Kafka
 - 9.5. Example of stream Processing with Kafka
- 10. Lab 5. Federated Learning
 - 10.1. Installation
 - 10.2. Working with Distributed FL environments.
 - 10.3. Hands on: Example of FL with existing models.
- 11. Lab 5. Kafka / MQTT / Ethereum / Federated Learning Student Project

6. Cronograma

6.1. Cronograma de la asignatura *

Sem	Actividad tipo 1	Actividad tipo 2	Tele-enseñanza	Actividades de evaluación
1	Tema 1 Introduction Duración: 02:30 LM: Actividad del tipo Lección Magistral	Lab 1. Linux Operating System Lab 2. Node-Red Programming Tool Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio	Lab 0.OOP with Java Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio Tutorías on-line de teoría y prácticas Duración: 02:30 OT: Otras actividades formativas / Evaluación	Prueba de nivelación de conocimientos de Programación Orientada a Objetos en Java EP: Técnica del tipo Examen de Prácticas Evaluación Progresiva Presencial Duración: 01:00
2	Tema 2. Indirect Communication. Publisher/ Subscriber Duración: 02:15 LM: Actividad del tipo Lección Magistral	Lab. 3. MQTT Protocol Duración: 02:15 PL: Actividad del tipo Prácticas de Laboratorio	Lab 0.OOP with Java Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio Tutorías on-line de teoría y prácticas Duración: 02:30 OT: Otras actividades formativas / Evaluación	
3	Tema 3. Stream Processing. Duración: 02:30 Duración: 02:30 LM: Actividad del tipo Lección Magistral	LAB 4. Stream Processing with Kafka. Kafka Command Interface (I) Duración: 02:15 PL: Actividad del tipo Prácticas de Laboratorio	LAB 4. Stream Processing with Kafka. Kafka Command Interface (I) Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio CT- AU- Programming with Blockchain Ethereum and Solidity Duración: 02:30 OT: Otras actividades formativas / Evaluación	Ex1 (Teoría + Prácticas)-Evaluación progresiva-MQTT. (RA62, RA63, RA64) EP: Técnica del tipo Examen de Prácticas Evaluación Progresiva Presencial Duración: 00:15
4	Tema 4. Blockchain Duración: 02:15 LM: Actividad del tipo Lección Magistral	LAB 4. Programming with the KAFKA JAVA API Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio	CT- AU- Programming with Blockchain Ethereum and Solidity Duración: 02:30 OT: Otras actividades formativas / Evaluación LAB 4. Programming with the KAFKA JAVA API Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio	
5	Tema 4. Blockchain Duración: 02:30 LM: Actividad del tipo Lección Magistral	Lab 5. MQTT/Kafka/Ethereum student project (I) Duración: 02:15 PL: Actividad del tipo Prácticas de Laboratorio	CT- AU- Programming with Blockchain Ethereum and Solidity Duración: 02:30 OT: Otras actividades formativas / Evaluación Lab 5. MQTT/Kafka/Ethereum student project (I) Duración: 02:30 PL: Actividad del tipo Prácticas de	

			Laboratorio	
6	Tema 5. Federated Learning Duración: 02:15 LM: Actividad del tipo Lección Magistral	Lab 5.Federated Learning Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio	CT- AU- Programming with Blockchain Ethereum and Solidity Duración: 02:30 OT: Otras actividades formativas / Evaluación Lab 5. MQTT/Kafka/Ethereum student project (I) Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio	Student Projects Oral Presentation(I) (RA64, RA66, RA67) PG: Técnica del tipo Presentación en Grupo Evaluación Progresiva Presencial Duración: 02:30 CT- Trabajo autónomo sobre Bockchain Ethereum (RA67, RA62) TI: Técnica del tipo Trabajo Individual Evaluación Progresiva No presencial Duración: 07:30
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				Ex2 (Teoría + Prácticas). (RA65-RA36, RA62-RA38, RA63-RA40) EX: Técnica del tipo Examen Escrito Evaluación Global Presencial Duración: 00:45 CT- Trabajo autónomo sobre Bockchain Ethereum (RA67, RA62) TI: Técnica del tipo Trabajo Individual Evaluación Global No presencial Duración: 07:30 Ex2 (Teoría + Prácticas). (RA65-RA36, RA62-RA38, RA63-RA40) EX: Técnica del tipo Examen Escrito Evaluación Progresiva Presencial Duración: 02:30 Ex1 (Teoría + Prácticas)-Evaluación progresiva-MQTT. (RA62, RA63, RA64) EX: Técnica del tipo Examen Escrito Evaluación Global Presencial Duración: 02:30

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
1	Prueba de nivelación de conocimientos de Programación Orientada a Objetos en Java	EP: Técnica del tipo Examen de Prácticas	Presencial	01:00	0%	8 / 10	CE07 CG12
3	Ex1 (Teoría + Prácticas)-Evaluación progresiva-MQTT. (RA62, RA63, RA64)	EP: Técnica del tipo Examen de Prácticas	Presencial	00:15	30%	4 / 10	CE07 CE03 CE04
6	Student Projects Oral Presentation(I) (RA64, RA66, RA67)	PG: Técnica del tipo Presentación en Grupo	Presencial	02:30	25%	0 / 10	CG11 CG12 CE03 CE04 CE06 CE07
6	CT- Trabajo autónomo sobre Bockchain Ethereum (RA67, RA62)	TI: Técnica del tipo Trabajo Individual	No Presencial	07:30	10%	0 / 10	CG11 CG12
17	Ex2 (Teoría + Prácticas). (RA65-RA36, RA62-RA38, RA63-RA40)	EX: Técnica del tipo Examen Escrito	Presencial	02:30	35%	4 / 10	CG10 CG12 CE03 CE04 CE06

7.1.2. Prueba evaluación global

Sem	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
17	Ex2 (Teoría + Prácticas). (RA65-RA36, RA62-RA38, RA63-RA40)	EX: Técnica del tipo Examen Escrito	Presencial	00:45	45%	0 / 10	CG10 CE03 CE04
17	CT- Trabajo autónomo sobre Bockchain Ethereum (RA67, RA62)	TI: Técnica del tipo Trabajo Individual	No Presencial	07:30	10%	0 / 10	CE07 CG11 CG12

17	Ex1 (Teoría + Prácticas)-Evaluación progresiva-MQTT. (RA62, RA63, RA64)	EX: Técnica del tipo Examen Escrito	Presencial	02:30	45%	4 / 10	CG10 CG12 CE03 CE04 CE06
----	--	-------------------------------------	------------	-------	-----	--------	--------------------------------------

7.1.3. Evaluación convocatoria extraordinaria

Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
Evaluación global de teoría y prácticas. (RA65-RA36, RA62-RA38, RA63-RA40)	EX: Técnica del tipo Examen Escrito	Presencial	01:00	100%	5 / 10	CG11 CG10 CG12 CE03 CE04 CE06 CE07

7.2. Criterios de evaluación

EVALUACIÓN PROGRESIVA:

Fecha1 (Evaluación Progresiva) --> 21 septiembre 2024:

NotaParcial1 = Ex1 (Teoría + Prácticas) * 0.3

Fecha2 (Evaluación Progresiva) --> 30 octubre 2024:

NotaParcial2 = Ex2 (Teoría + Prácticas) * 0.35

AND

IF **NotaParcial1**

NotaParcial1 = Ex1Recupera(Teoría + Prácticas) * 0.3

Fecha3 (Evaluación Progresiva) --> 30 octubre 2024:

NotaProyecto = proyecto*0.25 "

" Exposición oral del proyecto + Evaluación del profesor del trabajo hecho.

Nota definitiva:

IF **NotaParcial1** ≥ 4 AND **NotaParcial2** ≥ 4 :

Notadef = **NotaParcial1** + **NotaParcial2** + **NotaProyecto** + 0.1* Competencia Transversal.

ELSE:

Notadef = min (4.5, **Notadef**)

La asignatura se aprueba si **Notadef** ≥ 5 .

Explicación:

Los alumnos que opten por esta modalidad tendrán la posibilidad de presentar el primer parcial, que cubre teoría y prácticas. En la fecha final tendrán la posibilidad de recuperar este parcial si la nota obtenida es menos de 4. En la fecha final presentarán el segundo parcial de la asignatura, el cual tiene nota mínima de 4.

La nota definitiva se calcula como la suma de la nota de los dos parciales ponderada (si ambos superan el 4), la nota ponderada del proyecto y la nota ponderada de la competencia transversal. La asignatura se aprueba con la nota definitiva igual o mayor a 5.

EVALUACIÓN GLOBAL 30 octubre 2024:

NotaParcial1 = Ex1 (Teoría + Prácticas) * 0.45

NotaParcial2 = Ex2 (Teoría + Prácticas) * 0.45

Nota definitiva:

IF **NotaParcial1** ≥ 4 AND **NotaParcial2** ≥ 4 :

Notadef = **NotaParcial1** + **NotaParcial2** + 0.1* Competencia Transversal.

ELSE:

Notadef = min (4.5, **Notadef**)

La asignatura se aprueba si **Notadef** ≥ 5 .

Explicación:

Los alumnos que opten por esta opción realizarán los exámenes correspondientes a la primera y segunda parte del temario (teoría y prácticas). Cada examen vale un 45%, dando un total del 90%. El 10% restante se obtiene de la nota de la competencia transversal.

La nota definitiva se calcula como la suma de la nota de los dos parciales ponderada (si ambos superan el 4) y la nota ponderada de la competencia transversal. La asignatura se aprueba con la nota definitiva igual o mayor a 5.

CONVOCATORIA EXTRA-ORDINARIA

Notadef = Ex (Teoría + Prácticas)

La asignatura se aprueba si **Notadef** \geq 5.

8. Recursos didácticos

8.1. Recursos didácticos de la asignatura

Nombre	Tipo	Observaciones
Distributed Systems, concepts and design, 4th Edition. G. Coulouris. J. Dollimore, T. Kindberg, G. Blair. Addison Wesley, 2012.	Bibliografía	
Introduction to Reliable and Secure Distributed Programming Authors: Cachin, Christian, Guerraoui, Rachid, Rodrigues, Luís. Springer (2011)	Bibliografía	
Communication and Agreement Abstractions for Fault Tolerant Asynchronous Distributed Systems. Michel Raynal. Morgan & Claypool Publishers 2010	Bibliografía	
Making Sense of Stream Processing. By Martin Kleppmann Publisher: O'Reilly Released: May 2016	Bibliografía	
Apache Kafka Documentation	Recursos web	https://kafka.apache.org/
Moodle de la asignatura.	Recursos web	
Laboratorio de ordenadores con sistema operativo tipo Unix.	Equipamiento	
MQTT	Recursos web	mqtt.org https://www.hivemq.com/
Mastering Ethereum. Andreas Antonopoulos and Gavin Wood. Publisher: O'Reilly. 2018.	Bibliografía	
Fundamentos de Programación Orienta a Objetos	Recursos web	Curso on-line del MIT https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/

9. Otra información

9.1. Otra información sobre la asignatura

1. PRERREQUISITOS PREVIOS SOBRE DESTREZAS DE PROGRAMACIÓN ORIENTADA A OBJETOS.

Para todos aquellos alumnos que en cuya formación previa no figure formación básica demostrable en programación orientada a objetos en cualquier lenguaje, será necesario que, previamente al comienzo de curso, se formen en dicha destreza usando el lenguaje Java. El alumno es libre de formarse usando bibliografía básica sobre el tema o cursando cursos de plataformas de aprendizaje on-line como el curso del MIT:

<https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-092-introduction-to-programming-in-java-january-iap-2010/>

La profesora de prácticas de esta asignatura está disponible desde el día 6 de septiembre para resolver cualquier duda que surja sobre los conceptos estudiados. Por favor, contacte para ello con isabel.munoz@upm.es

La primera semana de impartición del curso, dichos alumnos deberán realizar una prueba de evaluación previa que figurará en la página de Moodle de la asignatura para que puedan presentarse al resto de pruebas.

2. COMPETENCIAS TRANSVERSALES

En esta asignatura se cubren las competencias genéricas de Resolución de Problemas (CG10) y Trabajo Autónomo (CG12)

RESOLUCIÓN DE PROBLEMAS (CG10)

En cuanto a la competencia de Resolución de Problemas (CG10) se trabaja todas las semanas durante las sesiones presenciales de teoría. La estructura de la clase teórica se desarrolla presentando primero el concepto teórico, para, a continuación, plantear a los alumnos pequeños problemas relacionados con dichos concepto teóricos.

TRABAJO AUTÓNOMO (CG12)

Respecto a la competencia transversal (CG12) sobre aprendizaje autónomo, se realiza sobre Blockchain Ethereum y de forma on-line. Esta competencia persigue que el alumno aprenda a instalar wallets y a diseñar y desplegar contratos. Como resultado del aprendizaje el alumno deberá entregar un guión para:

la Instalación de clientes remotos (wallets) y diseño, despliegue y uso de un nuevo contrato desde un wallet.

Adicionalmente la competencia (CG12) se trabaja de forma lateral en el proyecto de la asignatura. El

desarrollo de un mínimo proyecto requiere el aprendizaje de forma autónoma de herramientas de complejidad incluso alta. En cada sesión de prácticas se les da una orientación de cómo buscar documentación on-line de calidad de las herramientas software a utilizar en las prácticas presenciales

3. ACTIVIDADES SEMIPRESENCIALES

Para todos los alumnos se ofertan actividades semipresenciales relativas a prácticas, aprendizaje autónomo y tutorías on-line. Estas actividades aparecen planificadas como actividad de tele-enseñanza.

4. COMUNICACIÓN y PLATAFORMAS

La comunicación on-line con el alumno se realizará mediante la herramienta Teams institucional. Para su instalación y uso consulte <https://www.upm.es/UPM/ServiciosTecnologicos/Office365>.

Durante el horario de tutorías que se publique al comienzo de curso, el profesor estará disponible para su consulta presencial on-line. De cara a planificar las sesiones de tutorías, es imprescindible que el alumno solicite la tutoría con una antelación de al menos 24 h.

5. OBJETIVOS DE DESARROLLO SOSTENIBLE

La asignatura persigue dar una formación de calidad, inclusiva e igualitaria.

La asignatura describe cómo diseñar e implementar de manera eficiente sistemas distribuidos usando las tecnologías más avanzadas, para reducir el consumo de energía y contaminación que puedan provocar dichos sistemas.

Se promueve el uso de herramientas de Software libre.