ANX-PR/CL/001-01
LEARNING GUIDE

SUBJECT

593000506 - Distributed Systems For Iot

DEGREE PROGRAMME

59AH - Master Universitario En Internet Of Things (iot)

ACADEMIC YEAR & SEMESTER

2022/23 - Semester 1

Index

Learning guide

1. Description

1.1. Subject details

Name of the subject	593000506 - Distributed Systems For Iot
No of credits	4.5 ECTS
Type	Compulsory
Academic year ot the programme	First year
Semester of tuition	Semester 1
Tuition period	September-January
Tuition languages	English
Degree programme	59AH - Master Universitario en Internet Of Things (Iot)
Centre	59 - Escuela Tecnica Superior De Ingenieria Y Sistemas De Telecomunicacion
Academic year	2022-23

2. Faculty

2.1. Faculty members with subject teaching role

Name and surname	Office/Room	Email	Tutoring hours *
Sergio Arevalo Viñuales	4413	sergio.arevalo@upm.es	Sin horario.
Maria Isabel Muñoz Fernandez (Subject coordinator)	4412	isabel.munoz@upm.es	Sin horario.

* The tutoring schedule is indicative and subject to possible changes. Please check tutoring times with the faculty member in charge.

3. Prior knowledge recommended to take the subject

3.1. Recommended (passed) subjects

The subject - recommended (passed), are not defined.

3.2. Other recommended learning outcomes

- Programming languages, computer networks, operating systems

4. Skills and learning outcomes *

4.1. Skills to be learned

CB08 - Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios

CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

CE.05 - Diseñar y desarrollar sistemas distribuidos para dar soporte a aplicaciones IoT, evaluando las tecnologías mas apropiadas de acuerdo con los diferentes contextos de aplicación como son despositivos móviles, sistemas en tiempo real o sistemas ubícuos

CE.06 - Analizar el rendimiento, disponibilidad, escalabilidad y fiabilidad de los sistemas distribuidos empleados en aplicaciones IoT

CE.08 - Diseñar y desarrollar soluciones tecnológicas para implementar servicios IoT capaces de interactuar con diferentes fuentes de información y dispositivos distribuidos incluyendo el diseño de estructuras de intercambio de información eficientes

CG01 - Los alumnos demostrarán tener una visión del estado actual, las necesidades y los problemas que se plantean en el mundo de la IoT, así como de las arquitecturas y estándares más utilizados

CG02 - Los alumnos serán capaces de aplicar métodos y tecnologías avanzadas que les permitan abordar necesidades y problemas en aplicaciones IoT

CT.01 - Capacidad de uso de la lengua inglesa para el trabajo en contextos internacionales

4.2. Learning outcomes

RA12 - Coordinate IoT microservices using replicated state-machines services like Zookeeper.

RA10 - Design and implement a IoT microservice architecture based on Kafka stream processing platform.

RA11 - Choose the best replica consistency type for a IoT microservice.

RA13 - Apply CAP theorem principles to choose between availability and consistency

* The Learning Guides should reflect the Skills and Learning Outcomes in the same way as indicated in the Degree Verification Memory. For this reason, they have not been translated into English and appear in Spanish.

5. Brief description of the subject and syllabus

5.1. Brief description of the subject

Software that coordinates a set of connected computers in a communication network to get a certain goal is denominated Distributed System. This course will study different models of interactions among devices, computing services and data services used in IoT, Cloud Computing, Blockchain and Big Data. It will also study the scalability and availability techniques to replicate services with different degrees of consistency. First, typical Distributed System models and architectures are presented. Then, indirect communication publish/subscribe paradigm among processes of a distributed system will be explained, in order to introduce Stream Processing as a real time data processing. Next, Blockchain Ethereum architecture foundations will be shown for a non fault tolerant scenario (one server). The replica consistency and CAP theorem will be explained in order to introduce both Raft and the Fault Tolerant design of Blockchain-Ethereum. In the lab, we will develop distributed applications made up of hardware and software elements real and simulated with NodeRed. Then, the MQTT protocol will be described to use it along with NodeRED. Finally, we will move on to the Kafka Stream Processing tool, and we end with a sjhort example of Ethereum Blockchain.

5.2. Syllabus

1. Introduction
   1.1. Definition
   1.2. Abstractions
   1.3. Examples
   1.4. Models
2. Indirect Communication. Publisher-subscriber
   2.1. Definition
   2.2. Examples
   2.3. Properties
   2.4. Programming Model
   2.5. Implementation
3. Stream processing
   3.1. Case study definition
   3.2. Streaming 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. Non Fault Tolerant Approach (1)
   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. Replica consistency and CAP theorem
   5.1. Model

- 5.2. Atomic consistency
- 5.3. Sequential consistency
- 5.4. Causal Consistency
- 5.5. Eventual consistency
- 5.6. CAP theorem

6. Raft. Atomic consistency
   - 6.1. Introduction
   - 6.2. Model
   - 6.3. Raft algorithm basics
   - 6.4. Leader election
   - 6.5. Log replication
   - 6.6. Safety
7. 7 Blockchain. Ethereum. Fault Tolerant Approach (2)
   - 7.1. Peer-to-peer network
   - 7.2. Proof of work. Eventual consensus
   - 7.3. Eventual consistency replication
8. Lab 1. Linux Operating System
   - 8.1. Distributions
   - 8.2. Shell commands
9. Lab 2. NodeRED Programming Tool
   - 9.1. Programming Tool description
   - 9.2. Examples
10. Lab 3. MQTT Publish/Subscribe Protocol
   - 10.1. Protocol Description
   - 10.2. Quality of Service Semantics
   - 10.3. Example of stream Processing with Sensor Mobile, Node Red and MQTT
11. Lab 4. Stream Processing with Kafka
   - 11.1. Definition
   - 11.2. Installation

11.3. Kafka console commands

11.4. KAFKA Java API

11.5. Example

12. Lab 5. Blockchain with Ethereum

12.1. Definition

12.2. Example

6. Schedule

6.1. Subject schedule*

Week	Classroom activities	Laboratory activities	Distant / On-line	Assessment activities
1	**Lesson 1. Introduction** Duration: 02:45	**Lab 1. Linux. Node Red** Duration: 02:45		
2	**Lesson 2. Indirect communication. Publisher-subscriber. Lesson 3. Stream Processing** Duration: 01:45	**Lab 2. MQTT Protocol** Duration: 01:45		
3	**Lesson 3. Stream Processing Lesson 4. Blockchain. Ethereum. Non Fault Tolerant Approach (1)** Duration: 01:45	**Lab3. MQTT Protocol** Duration: 01:45		
4	**Lesson 4. Blockchain. Ethereum. Non Fault Tolerant Approach (1)** Duration: 01:30	**Lab3. MQTT Protocol. Kafka Stream Processing (1h + 0,45m)** Duration: 01:45		**Theory middle term exam 1.** Continuous assessment Presential Duration: 00:15
5	**Lesson 4. Blockchain. Ethereum. Non Fault Tolerant Approach (1)** Duration: 02:45	**Lab 4. Kafka Stream Processing** Duration: 02:30		**LAB-EX-1 MQTT MId Term Exam** Continuous assessment Presential Duration: 00:15
6	**Lesson 5.Replica consistency and CAP theorem** Duration: 01:30	**Lab 4. Kafka Stream Processing** Duration: 01:45		**Theory middle term exam 2** Continuous assessment Presential Duration: 00:15
7	**Lesson 6. Raft** Duration: 01:45	**Lab 5. Ethereum Blockchain** Duration: 01:30		**LAB-EX 2- KAFKA MId Term Exam** Continuous assessment Presential Duration: 00:15
8	**Lesson 6. Raft Lesson 7. Blockchain. Ethereum. Fault Tolerant Approach (2)** Duration: 02:00	**Lab 6. Student Project MQTT/Kafka/Ethereum** Duration: 01:15		**LAB-EX 3- Blockchain with Ethereum. MId Term Exam** Continuous assessment Presential Duration: 00:15
9	**Lesson 7. Blockchain. Ethereum. Fault Tolerant Approach (2)** Duration: 03:30	**Studend Project presentation (I)** Duration: 02:00		

10				**Project MQTT/Kafka/Ethereum presentation** Continuous assessment Presential Duration: 03:15 **Theory middle term exam 3** Continuous assessment Presential Duration: 00:15
11				
12				
13				
14				
15				
16				
17				**Global Theory final exam.** Final examination Presential Duration: 00:45 **Project MQTT/Kafka/Ethereum presentation** Final examination Presential Duration: 02:00 **Global Lab final exam.** Final examination Presential Duration: 00:45

Depending on the programme study plan, total values will be calculated according to the ECTS credit unit as 26/27 hours of student face-to-face contact and independent study time.

* The schedule is based on an a priori planning of the subject; it might be modified during the academic year, especially considering the COVID19 evolution.

7. Activities and assessment criteria

7.1. Assessment activities

7.1.1. Assessment

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
4	Theory middle term exam 1.		Face-to-face	00:15	15%	0 / 10	CB10 CE.08 CT.01 CE.05
5	LAB-EX-1 MQTT MId Term Exam		Face-to-face	00:15	10%	0 / 10	CG01 CG02 CE.08 CT.01
6	Theory middle term exam 2		Face-to-face	00:15	15%	0 / 10	CB08 CT.01 CE.05
7	LAB-EX 2- KAFKA MId Term Exam		Face-to-face	00:15	10%	0 / 10	CE.06 CG01 CE.08 CT.01 CE.05
8	LAB-EX 3- Blockchain with Ethereum. MId Term Exam		Face-to-face	00:15	5%	0 / 10	CE.06 CE.08 CT.01 CE.05
10	Project MQTT/Kafka/Ethereum presentation		Face-to-face	03:15	25%	0 / 10	CB10 CE.06 CG01 CG02 CB08 CE.08 CT.01 CE.05
10	Theory middle term exam 3		Face-to-face	00:15	20%	0 / 10	CG01 CG02 CE.08 CT.01 CE.05

7.1.2. Global examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
17	Global Theory final exam.		Face-to-face	00:45	50%	0 / 10	CB10 CE.06 CG01 CG02 CE.08 CT.01 CE.05
17	Project MQTT/Kafka/Ethereum presentation		Face-to-face	02:00	25%	5 / 10	CT.01 CE.05 CB10 CE.06 CG01 CG02 CB08 CE.08
17	Global Lab final exam.		Face-to-face	00:45	25%	0 / 10	CG01 CG02 CE.08 CT.01 CE.05 CB10 CE.06

7.1.3. Referred (re-sit) examination

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
Theory final exam		Face-to-face	01:00	50%	0 / 10	CT.01 CE.05 CB10 CE.06 CG01 CG02 CE.08
Project MQTT/Kafka/Ethereum presentation		Face-to-face	01:00	25%	5 / 10	CT.01 CE.05 CB10 CE.06 CG01 CG02 CB08 CE.08

Lab Exam		Face-to-face	01:00	25%	0 / 10	CB10 CE.06 CG01 CG02 CE.08 CT.01 CE.05

7.2. Assessment criteria

Course evaluation on Regular period.

a) On-going evaluation system.

Block I. Theory. 3 individual written exams (50%)

Block II. Laboratory (50%) that consists of:

- 3 individual written exams (25 %)

- 1 Group project (25%)

- In case the student fails the on-going evaluation but gets 4 or more in one of the blocks he/she will be able to do only the failed block in the global exam and/or in the extra period exam in July.

b) Global evaluation System.

The students can follow this global evaluation system. One exam per block will be done at the end of the course.

In case the student fails the global exam evaluation but gets 4 or more in one of the blocks he/she will be able to do only the failed block in the extra period exam in July.

c) Course evaluation on Extra period (July exam).

Students will do one exam per failed block with less than 4 points out of 10.

The algorithm to obtain the final grade of the course is the following:

FinalGrade = theoryGrade * 0.5 + LabGrade *0-25 + Project *0.25

8. Teaching resources

8.1. Teaching resources for the subject

Name	Type	Notes
Distributed Systems, concepts and design, 4th Edition. G. Coulouris. J. Dollimore, T. Kindberg, G. Blair. Addison Wesley, 2012.	Bibliography	
Introduction to Reliable and Secure Distributed Programming Authors: Cachin, Christian, Guerraoui, Rachid, Rodrigues, Luís. Springer (2011)	Bibliography	
Communication and Agreement Abstractions for Fault Tolerant Asynchronous Distributed Systems. Michel Raynal. Morgan & Claypool Publishers 2010	Bibliography	

Making Sense of Stream Processing. By Martin Kleppmann Publisher: O'Reilly Released: May 2016	Bibliography	
Raft Paper	Bibliography	D. Ongaro, J. Ousterhout. In Search of an Understandable Consensus Algorithm. USENIX Annual Technical Conference (ATC), Philadelphia, PA, 2014
Apache Kafka Documentation	Web resource	https://kafka.apache.org/
Moodle de la asignatura	Web resource	moodle upm
Laboratorio de ordenadores con sistema operativo tipo Unix.	Equipment	
MQTT	Web resource	http://mqtt.org/ MQTT is a machine-to-machine (M2M)/"Internet of Things" connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It
Zookeeper	Web resource	https://zookeeper.apache.org/
Mastering Ethereum. Andreas Publisher: Antonopoulos O'Reilly. and Gavin 2018. Wood.	Bibliography	
Blockchain Paper	Bibliography	From blockchain consensus back to Byzantine consensus. Vincent Gramoli. Future Generation Computer Systems, V107, June 2020.

9. Other information

9.1. Other information about the subject

UPM Teams Tool will be used as the default communication tool with the student. Please, check

https://www.upm.es/UPM/ServiciosTecnologicos/Office365.