ANX-PR/CL/001-01

LEARNING GUIDE

SUBJECT

105000443 - Programming Scalable Systems

DEGREE PROGRAMME

10II - Grado En Ingenieria Informatica

ACADEMIC YEAR & SEMESTER

2023/24 - Semester 2

Index

Learning guide

1. Description

1.1. Subject details

Name of the subject	105000443 - Programming Scalable Systems
No of credits	3 ECTS
Type	Optional
Academic year ot the programme	Fourth year
Semester of tuition	Semester 8
Tuition period	February-June
Tuition languages	English
Degree programme	10II - Grado en Ingenieria Informatica
Centre	10 - Escuela Tecnica Superior De Ingenieros Informaticos
Academic year	2023-24

2. Faculty

2.1. Faculty members with subject teaching role

Name and surname	Office/Room	Email	Tutoring hours *
Julio Mariño Carballo	2310	julio.marino@upm.es	Sin horario. Publication pending
Angel Herranz Nieva (Subject coordinator)	2309	angel.herranz@upm.es	Sin horario. Publication pending
Lars-ake Fredlund	2309	larsake.fredlund@upm.es	Sin horario. Publication pending

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

2.2. Research assistants

Name and surname	Email	Faculty member in charge
Bueso De Barrio, Luis Eduardo	luiseduardo.bueso.debarrio@upm.es	Herranz Nieva, Angel

3. Prior knowledge recommended to take the subject

3.1. Recommended (passed) subjects

- Concurrencia

- Sistemas Operativos

3.2. Other recommended learning outcomes

- Functional Programming

4. Skills and learning outcomes *

4.1. Skills to be learned

CG-13/CE55 - Capacidad de comunicarse de forma efectiva con los compañeros, usuarios (potenciales) y el público en general acerca de cuestiones reales y problemas relacionados con la especialización elegida.

CG-6 - Capacidad de abstracción, análisis y síntesis

Ce 12/16 - Conocer los campos de aplicación de la informática, y tener una apreciación de la necesidad de poseer unos conocimientos técnicos profundos en ciertas áreas de aplicación; apreciación del grado de esta necesidad en, por lo menos, una situación.

Ce 17 - Conocer los temas informáticos avanzados de modo que permita a los alumnos vislumbrar y entender las fronteras de la disciplina, por medio de la inclusión de experiencias de aprendizaje que dirigen a los alumnos desde los temas elementales a los temas avanzados o los temas de los que se nutren los novísimos desarrollos.

Ce 19/20 - Conocimiento de los tipos apropiados de soluciones, y comprensión de la complejidad de los problemas informáticos y la viabilidad de su solución.

Ce 44 - Conocimiento de tecnologías punteras relevantes y su aplicación.

4.2. Learning outcomes

RA542 - Entender las fortalezas del ecosistema Erlang/OTP y sus características para crear sistemas de alta disponibilidad, fiables, escalables, y mantenibles.

RA543 - Conocimientos de programación funcional.

RA276 - Dado un campo de aplicación de la informática, evaluar y diseñar el sistema informático más apropiado para resolver alguno de sus problemas, exponiendo las dificultades técnicas y los limites de la aplicación.

* 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

The development of new applications with a strong presence on the Internet, such as social network applications, present great challenges:

- Design and programming of back-ends (services) that support the popularization of the application.
- Design and programming of front-ends (mobile applications and web applications) with good user experience, elegant interfaces, and efficient access to the back-end.

This course **focuses** on the design and **programming the back-end** of these types of applications using **Elixir** functional programming langauage and the Erlang/OTP ecosystem.

The fundamental requirements behind these services are

- Availability (they cannot stop answering requests).
- Reliability (must be fault tolerant).
- Scalability (they must be able to serve huge numbers of simultaneous requests).
- Maintainability (they must be easily adaptable to new requirements).
- Accessibility (they must define well-structured APIs to serve the front-end).

The course delves into the Erlang/OTP ecosystem and the Elixir programming language. Elixir leverages Erlang's virtual machine (BEAM), well known for its ability to run **low-latency, distributed and fault-tolerant systems**, to develop back-ends with the aforementioned features (available, reliable and scalable). Elixir is a **modern functional programming language** designed for productivity, with well documented libraries and frameworks and with strong metaprogramming capabilities (maintainable and accesible).

Some examples of applications and systems where these languages have been used are: WhatsApp, Discord, Cabify, bet365, Nintendo Switch multi-user online gaming or RabbitMQ.

5.2. Syllabus

1. Introduction to Elixir

2. Functional Programming

3. Concurrency in Elixir

4. Distribution in Elixir

5. Elixir Applications and OTP (Open Telecom Plantform)

6. Libraries to Scale

7. Project

6. Schedule

6.1. Subject schedule*

Week	Classroom activities	Laboratory activities	Distant / On-line	Assessment activities
1	**Get ready for Elixir** Duration: 02:00 Lecture			
2	**Discussion about Proposed Exercises** Duration: 00:15 Problem-solving class **A New Language 1/2** Duration: 01:45 Lecture			**Proposed Exercises** Individual work Continuous assessment Not Presential Duration: 03:00
3	**Discussion about Proposed Exercises** Duration: 00:15 Problem-solving class **A New Language 2/2** Duration: 01:45 Lecture			**Proposed Exercises** Individual work Continuous assessment Not Presential Duration: 03:00
4	**Discussion about Proposed Exercises** Duration: 00:15 Problem-solving class **Functional Programming 1/3** Duration: 01:45 Lecture			**Proposed Exercises** Individual work Continuous assessment Not Presential Duration: 03:00
5	**Discussion about Proposed Exercises** Duration: 00:15 Problem-solving class **Functional Programming 2/3** Duration: 01:45 Lecture			**Proposed Exercises** Individual work Continuous assessment Not Presential Duration: 03:00
6	**Discussion about Proposed Exercises** Duration: 00:15 Problem-solving class **Functional Programming 3/3** Duration: 01:45 Lecture			**Proposed Exercises** Individual work Continuous assessment Not Presential Duration: 03:00
7	**Discussion about Proposed Exercises** Duration: 00:15 Problem-solving class **Concurrency in Elixir 1/2** Duration: 01:45 Lecture			**Proposed Exercises** Individual work Continuous assessment Not Presential Duration: 03:00

8	**Discussion about Proposed Exercises** Duration: 00:15 Problem-solving class **Concurrency in Elixir 2/2** Duration: 01:45 Lecture			**Proposed Exercises** Individual work Continuous assessment Not Presential Duration: 03:00
9	**Discussion about Proposed Exercises** Duration: 00:15 Problem-solving class **Distribution in Elixir** Duration: 01:45 Lecture			**Proposed Exercises** Individual work Continuous assessment Not Presential Duration: 03:00
10	**Discussion about Proposed Exercises** Duration: 00:15 Problem-solving class **Elixir Applications and OTP (Open Telecom Plantform)** Duration: 01:45 Lecture			**Proposed Exercises** Individual work Continuous assessment Not Presential Duration: 03:00
11	**Project Proposals and Frameworks** Duration: 02:00 Lecture			**Mandatory Exercises Submission (ME)** Individual work Continuous assessment and final examination Not Presential Duration: 00:10
12	**Libraries to Scale** Duration: 02:00 Lecture			**Project Development** Group work Continuous assessment Not Presential Duration: 03:00
13	**Talk from Industry** Duration: 02:00 Additional activities			**Project Development** Group work Continuous assessment Not Presential Duration: 03:00
14				**Project Development** Group work Continuous assessment Not Presential Duration: 03:00
15	**Discussion about Project Development** Duration: 02:00 Problem-solving class			**Project Development** Group work Continuous assessment Not Presential Duration: 03:00
16				**Project Submission (PD)** Group work Continuous assessment and final examination Not Presential Duration: 00:10

17				**Project Presentation (PP)** Group presentation Continuous assessment and final examination Presential Duration: 00:15

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
2	Proposed Exercises	Individual work	No Presential	03:00	0%	0 / 10	
3	Proposed Exercises	Individual work	No Presential	03:00	0%	0 / 10	
4	Proposed Exercises	Individual work	No Presential	03:00	0%	0 / 10	
5	Proposed Exercises	Individual work	No Presential	03:00	0%	0 / 10	
6	Proposed Exercises	Individual work	No Presential	03:00	0%	0 / 10	
7	Proposed Exercises	Individual work	No Presential	03:00	0%	0 / 10	
8	Proposed Exercises	Individual work	No Presential	03:00	0%	0 / 10	
9	Proposed Exercises	Individual work	No Presential	03:00	0%	0 / 10	
10	Proposed Exercises	Individual work	No Presential	03:00	0%	0 / 10	
11	Mandatory Exercises Submission (ME)	Individual work	No Presential	00:10	20%	0 / 10	
12	Project Development	Group work	No Presential	03:00	0%	0 / 10	
13	Project Development	Group work	No Presential	03:00	0%	0 / 10	
14	Project Development	Group work	No Presential	03:00	0%	0 / 10	
15	Project Development	Group work	No Presential	03:00	0%	0 / 10	
16	Project Submission (PD)	Group work	No Presential	00:10	50%	0 / 10	CG-6 CG-13/CE55 Ce 12/16 Ce 17 Ce 19/20 Ce 44
17	Project Presentation (PP)	Group presentation	Face-to-face	00:15	30%	0 / 10	CG-6 CG-13/CE55 Ce 12/16 Ce 17 Ce 19/20 Ce 44

7.1.2. Global examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
11	Mandatory Exercises Submission (ME)	Individual work	No Presential	00:10	20%	0 / 10	
16	Project Submission (PD)	Group work	No Presential	00:10	50%	0 / 10	CG-6 CG-13/CE55 Ce 12/16 Ce 17 Ce 19/20 Ce 44
17	Project Presentation (PP)	Group presentation	Face-to-face	00:15	30%	0 / 10	CG-6 CG-13/CE55 Ce 12/16 Ce 17 Ce 19/20 Ce 44

7.1.3. Referred (re-sit) examination

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
Mandatory Exercise Submission (ME)	Individual work	Face-to-face	00:10	20%	0 / 10	CG-6 Ce 19/20 Ce 44
Project Submission (PD)	Group work	Face-to-face	00:10	50%	0 / 10	CG-6 CG-13/CE55 Ce 12/16 Ce 17 Ce 19/20 Ce 44
Project Presentation (PP)	Group presentation	Face-to-face	00:15	30%	0 / 10	CG-6 CG-13/CE55 Ce 12/16 Ce 17 Ce 19/20 Ce 44

7.2. Assessment criteria

Assesment

The course has several evaluable parts (that are **preserved just during the the semester**).

- **Voluntary proposed exercises (VE 0%)**: several exercise sheets will be published every two or three weeks. The submission of these exercises is voluntary and will not be graded,
- **Mandatory final exercises (ME 20%, non-recoverable)**: a selection of the proposed exercises will be required to submit. These exercises are considered non-recoverable because the knowledge they assess is essential to successfully address the project (PD).
- **Mandatory Project development (PD 50%, non-recoverable)**: in teams of two or three, the students have to develop a project (and defend it in a final presentation). The development will be done under a version control system (Git) in order to justify the implication of every student. Non-recoverable due to lack of time.
- **Mandatory Presentation of the project development (PP 30%, non-recoverable)**: in a final presentation will be done by every team in order to explain and defend its decisions. Non-recoverable due to lack of time.

Assesment Formula: 0.2 * ME + 0.5 *PD + 0.3 PP

Progressive Assesment

During the progressive assesment the student has the oportunity to discuss about the proposed exercises in every class.

Final grade will be calculated with the **assesment formula**.

Global Assesment

Final grade will be calculated with the **assesment formula**.

Extraordinay Period Assesment

For the extraordinary period assesment the student will be required to submit the mandatory exercises (ME), to submit a project development (PD) and to do a project presentation (PP). Mandatory exercise sheet will be publish 10 days before the exam day. Project will be selected by the student 10 days before the exam day.

Final grade will be calculated with the **assesment formula**.

Academic Fraud

Any type of fraudulent behavior such as copying of exercises and practical developments will be prosecuted. The students involved (copiers and willing copied) will be subject to the current regulations of the UPM in this regard. Main implication is that the student **will not pass the course** in the semester. The matter will be formaly notified to the institution in order to take extra punitive actions.

8. Teaching resources

8.1. Teaching resources for the subject

Name	Type	Notes
Introducing Elixir. Simon St. Laurent and J. David Eisenberg (O'Reilly, 2014).	Bibliography	class book
Études for Elixir. J. David Eisenberg (O'Reilly Media, 2013)..	Bibliography	exercises book
Elixir School (https://elixirschool.com/en/)	Web resource	
Elements of Functional Programming. Chris Reade (Addison-Wesley Longman Publishing Co., 1989.)	Bibliography	