



POLITÉCNICA

INTERNATIONAL  
CAMPUS OF  
EXCELLENCE

COORDINATION PROCESS OF  
LEARNING ACTIVITIES  
PR/CL/001



E.T.S. de Ingenieros  
Informaticos

# ANX-PR/CL/001-01

## LEARNING GUIDE

### SUBJECT

**105000443 - Programming Scalable Systems**

### DEGREE PROGRAMME

10II - Grado En Ingenieria Informatica

### ACADEMIC YEAR & SEMESTER

2021/22 - Semester 2

## Index

---

### Learning guide

1. Description.....	1
2. Faculty.....	1
3. Prior knowledge recommended to take the subject.....	2
4. Skills and learning outcomes .....	2
5. Brief description of the subject and syllabus.....	3
6. Schedule.....	5
7. Activities and assessment criteria.....	8
8. Teaching resources.....	12

## 1. Description

---

### 1.1. Subject details

<b>Name of the subject</b>	105000443 - Programming Scalable Systems
<b>No of credits</b>	3 ECTS
<b>Type</b>	Optional
<b>Academic year of the programme</b>	Fourth year
<b>Semester of tuition</b>	Semester 8
<b>Tuition period</b>	February-June
<b>Tuition languages</b>	English
<b>Degree programme</b>	10II - Grado en Ingenieria Informatica
<b>Centre</b>	10 - Escuela Tecnica Superior De Ingenieros Informaticos
<b>Academic year</b>	2021-22

## 2. Faculty

---

### 2.1. Faculty members with subject teaching role

<b>Name and surname</b>	<b>Office/Room</b>	<b>Email</b>	<b>Tutoring hours *</b>
Julio Mariño Carballo	2310	julio.marino@upm.es	Sin horario. Pendiente de publicación
Angel Herranz Nieva (Subject coordinator)	2309	angel.herranz@upm.es	Sin horario. Pendiente de publicación

Clara Benac Earle	2316	clara.benac@upm.es	Sin horario. Pendiente de su publicación
Lars-ake Fredlund	2309	larsake.fredlund@upm.es	Sin horario. Pendiente de publicación

\* 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

- 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 límites 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 language 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 and Distribution
4. Elixir Applications and OTP (Open Telecom Plantform)
5. Advanced Libraries
6. Project

## 6. Schedule

### 6.1. Subject schedule\*

Week	Face-to-face classroom activities	Face-to-face laboratory activities	Distant / On-line	Assessment activities
1	<b>Arrancando con Elixir</b> Duration: 02:00 Lecture			<b>Proposed Exercises</b> Individual work Continuous assessment Not Presential Duration: 03:00
2	<b>Discussion about Proposed Exercises</b> Duration: 00:15 Problem-solving class  <b>A New Language 1/2</b> Duration: 01:45 Lecture			<b>Proposed Exercises</b> Individual work Continuous assessment Not Presential Duration: 03:00
3	<b>Discussion about Proposed Exercises</b> Duration: 00:15 Problem-solving class  <b>A New Language 2/2</b> Duration: 01:45 Lecture			<b>Proposed Exercises</b> Individual work Continuous assessment Not Presential Duration: 03:00
4	<b>Discussion about Proposed Exercises</b> Duration: 00:15 Problem-solving class  <b>Functional Programming 1/2</b> Duration: 01:45 Lecture			<b>Proposed Exercises</b> Individual work Continuous assessment Not Presential Duration: 03:00
5	<b>Discussion about Proposed Exercises</b> Duration: 00:15 Problem-solving class  <b>Functional Programming 2/2</b> Duration: 01:45 Lecture			<b>Proposed Exercises</b> Individual work Continuous assessment Not Presential Duration: 03:00
6	<b>Discussion about Proposed Exercises</b> Duration: 00:15 Problem-solving class  <b>Concurrency and Distribution 1/2</b> Duration: 01:45 Lecture			<b>Proposed Exercises</b> Individual work Continuous assessment Not Presential Duration: 03:00
7	<b>Discussion about Proposed Exercises</b> Duration: 00:15 Problem-solving class  <b>Concurrency and Distribution 2/2</b> Duration: 01:45 Lecture			<b>Proposed Exercises</b> Individual work Continuous assessment Not Presential Duration: 03:00

8	<p><b>Discussion about Proposed Exercises</b> Duration: 00:15 Problem-solving class</p> <p><b>Elixir Applications and OTP 1/2</b> Duration: 01:45 Lecture</p>			<p><b>Proposed Exercises</b> Individual work Continuous assessment Not Presential Duration: 03:00</p>
9	<p><b>Discussion about Proposed Exercises</b> Duration: 00:15 Problem-solving class</p> <p><b>Elixir Applications and OTP 2/2</b> Duration: 01:45 Lecture</p>			<p><b>Proposed Exercises</b> Individual work Continuous assessment Not Presential Duration: 03:00</p>
10	<p><b>Advanced Libraries</b> Duration: 02:00 Lecture</p>			
11	<p><b>Project Proposals</b> Duration: 02:00 Additional activities</p>			<p><b>Final Proposed Exercises</b> Individual work Continuous assessment Not Presential Duration: 03:00</p>
12	<p><b>Discussion about Project Proposals</b> Duration: 02:00 Problem-solving class</p>			<p><b>Project Development</b> Group work Continuous assessment Not Presential Duration: 03:00</p>
13				<p><b>Project Development</b> Group work Continuous assessment Not Presential Duration: 03:00</p>
14				<p><b>Project Development</b> Group work Continuous assessment Not Presential Duration: 03:00</p>
15				<p><b>Project Development</b> Group work Continuous assessment Not Presential Duration: 03:00</p>
16				<p><b>Project Development</b> Group work Continuous assessment Not Presential Duration: 03:00</p>
17				<p><b>Project Presentation</b> Group presentation Continuous assessment Presential Duration: 00:30</p> <p><b>Examen para la modalidad "sólo examen final"</b> Other assessment Final examination Not Presential Duration: 05:00</p>



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. Continuous assessment

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
1	Proposed Exercises	Individual work	No Presential	03:00	2%	/ 10	CG-6 Ce 19/20 Ce 44
2	Proposed Exercises	Individual work	No Presential	03:00	2%	/ 10	CG-6 Ce 19/20 Ce 44
3	Proposed Exercises	Individual work	No Presential	03:00	2%	/ 10	CG-6 Ce 19/20 Ce 44
4	Proposed Exercises	Individual work	No Presential	03:00	2%	/ 10	CG-6 Ce 19/20 Ce 44
5	Proposed Exercises	Individual work	No Presential	03:00	2%	/ 10	CG-6 Ce 19/20 Ce 44
6	Proposed Exercises	Individual work	No Presential	03:00	2%	/ 10	CG-6 Ce 19/20 Ce 44
7	Proposed Exercises	Individual work	No Presential	03:00	2%	/ 10	CG-6 Ce 19/20 Ce 44
8	Proposed Exercises	Individual work	No Presential	03:00	2%	/ 10	CG-6 Ce 19/20 Ce 44
9	Proposed Exercises	Individual work	No Presential	03:00	2%	/ 10	CG-6 Ce 19/20 Ce 44
11	Final Proposed Exercises	Individual work	No Presential	03:00	2%	/ 10	
12	Project Development	Group work	No Presential	03:00	10%	/ 10	Ce 12/16 Ce 17 Ce 19/20 Ce 44 CG-6 CG-13/CE55

13	Project Development	Group work	No Presential	03:00	10%	/ 10	Ce 12/16 Ce 17 Ce 19/20 Ce 44 CG-6 CG-13/CE55
14	Project Development	Group work	No Presential	03:00	10%	/ 10	Ce 12/16 Ce 17 Ce 19/20 Ce 44 CG-6 CG-13/CE55
15	Project Development	Group work	No Presential	03:00	10%	/ 10	Ce 12/16 Ce 17 Ce 19/20 Ce 44 CG-6 CG-13/CE55
16	Project Development	Group work	No Presential	03:00	10%	/ 10	Ce 12/16 Ce 17 Ce 19/20 Ce 44 CG-6 CG-13/CE55
17	Project Presentation	Group presentation	Face-to-face	00:30	30%	/ 10	CG-6 CG-13/CE55 Ce 12/16 Ce 17 Ce 19/20 Ce 44

### 7.1.2. Final examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
17	Examen para la modalidad "sólo examen final"	Other assessment	No Presential	05:00	100%	/ 10	Ce 12/16 Ce 19/20 Ce 44 CG-6 Ce 17

### 7.1.3. Referred (re-sit) examination

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
Proposed Exercises	Individual work	Face-to-face	01:00	20%	/ 10	Ce 19/20 Ce 44 CG-6
Project Development	Group work	Face-to-face	15:00	50%	/ 10	Ce 19/20 Ce 44 CG-6 Ce 17 Ce 12/16 CG-13/CE55
Project Presentation	Group presentation	Face-to-face	00:30	30%	/ 10	CG-13/CE55 Ce 19/20 Ce 44 CG-6 Ce 17 Ce 12/16

## 7.2. Assessment criteria

La asignatura tiene tres partes evaluables:

- Ejercicios propuestos.
- Desarrollo práctico: en grupos los alumnos harán un desarrollo que tendrán que presentar y defender, dicho desarrollo se realizará bajo sistemas de control de versiones que permitan justificar el trabajo realizado por cada estudiante.
- Presentación del desarrollo práctico: sesión oral presencial o telepresencial para defender el desarrollo realizado.

Los porcentajes son los indicados en los criterios de evaluación exigiéndose una nota mínima de 4 sobre 10 en cada una de las partes.

### Evaluación en modo "sólo examen final"

Para aquellos alumnos que soliciten evaluación de la asignatura en modo "sólo examen final" en los plazos establecidos por la Universidad, se realizará un examen final en el que se evaluarán todos los bloques arriba mencionados, y que constará de:

- Hoja de ejercicios y desarrollo práctico a realizar ante un ordenador

La solicitud de esta modalidad se hará mediante escrito dirigido al coordinador de la asignatura que se presentará en registro de la Facultad en los 15 días posteriores al inicio de las clases. El escrito se ajustará a este formato:

D. \_\_\_\_\_ con DNI \_\_\_\_\_ y nº de matrícula \_\_\_\_\_

SOLICITA:

Ser evaluado en este semestre mediante el sistema de evaluación mediante sólo prueba final establecido por las siguientes asignaturas:

- Asignatura \_\_\_\_\_ titulación \_\_\_\_\_ curso \_\_\_\_\_

Firmado:

## Comportamiento fraudulento

Se perseguirá todo tipo de comportamiento fraudulento como la copia de ejercicios y desarrollos prácticos. A los alumnos implicados (copiadores y copiados anuentes) se les aplicará la normativa vigente de la UPM al respecto. El asunto será notificado a Jefatura de Estudios y a los tutores curriculares de los alumnos. Se prevé el uso de exámenes a una selección de grupos como medio de lucha contra las copias.

## 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
Phoenix Framework ( <a href="https://www.phoenixframework.org">https://www.phoenixframework.org</a> )	Web resource	
Elixir School ( <a href="https://elixirschool.com/en/">https://elixirschool.com/en/</a> )	Web resource	
Elements of Functional Programming. Chris Reade (Addison-Wesley Longman Publishing Co., 1989.)	Bibliography	
Structure and Interpretation of Computer Programs. Harold Abelson, Gerald Jay Sussman, Julie Sussman (MIT Press, 1985)	Bibliography	