



POLITÉCNICA

INTERNATIONAL  
CAMPUS OF  
EXCELLENCE

COORDINATION PROCESS OF  
LEARNING ACTIVITIES  
PR/CL/001



E.T.S. de Ingenieros  
Informáticos

# ANX-PR/CL/001-01

## LEARNING GUIDE

### SUBJECT

**105000394 - Programming Project**

### 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.....	7
8. Teaching resources.....	10

## 1. Description

---

### 1.1. Subject details

<b>Name of the subject</b>	105000394 - Programming Project
<b>No of credits</b>	3 ECTS
<b>Type</b>	Optional
<b>Academic year of the programme</b>	Third year
<b>Semester of tuition</b>	Semester 6
<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>
Guillermo Roman Diez (Subject coordinator)	2304	guillermo.roman@upm.es	M - 12:00 - 15:00 W - 12:00 - 15:00 Please send an e-mail to set up an appointment
Raul Alonso Calvo	2315	raul.alonso@upm.es	M - 10:00 - 13:00 W - 10:00 - 13:00 Please send an e-mail to set up an appointment

\* 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

- Programacion Ii
- Algoritmos Y Estructura De Datos
- Programacion I
- Programacion Para Sistemas
- Concurrencia

#### 3.2. Other recommended learning outcomes

The subject - other recommended learning outcomes, are not defined.

### 4. Skills and learning outcomes \*

---

#### 4.1. Skills to be learned

CG-1/21 - Capacidad de resolución de problemas aplicando conocimientos de matemáticas, ciencias e ingeniería.

CG-19 - Capacidad de usar las tecnologías de la información y la comunicación.

CG-2/CE45 - Capacidad para el aprendizaje autónomo y la actualización de conocimientos, y reconocimiento de su necesidad en el área de la informática.

CG-24/25/26/27 - Capacidad para trabajar en el contexto internacional, comunicándose en lengua inglesa y adaptándose a un nuevo entorno.

CG-3/4 - Saber trabajar en situaciones carentes de información y bajo presión, teniendo nuevas ideas, siendo creativo.

CG-5 - Capacidad de gestión de la información.

CG-6 - Capacidad de abstracción, análisis y síntesis

CG-7:10/16/17 - Capacidad para trabajar dentro de un equipo, organizando, planificando, tomando decisiones, negociando y resolviendo conflictos, relacionándose, y criticando y haciendo autocrítica

Ce 14/15 - Conocer el software, el hardware y las aplicaciones existentes en el mercado, así como el uso de sus elementos, y capacidad para familiarizarse con nuevas aplicaciones informáticas.

## 4.2. Learning outcomes

RA283 - Experiencia del desempeño profesional del ingeniero y de sus funciones más habituales en un entorno real de empresa.

RA284 - Capacitación para diseñar las líneas maestras de un proyecto.

RA521 - Resolver problemas algorítmicos no triviales Documentar clases y bibliotecas, tanto de manera pública

RA522 - Usar y definir estructuras de datos eficientes y adecuadas a cada problema

RA278 - Desarrollar la solución matemática y algorítmica mas apropiada a un problema informático que requiera un tratamiento especialmente complejo, analizando y exponiendo su viabilidad.

\* 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

This course aims at putting in practice the knowledge acquired by the students during the programming courses by means of the development of a programming project. The main goal of this course is to familiarize the students with the professional software development: the student will take design and programming decisions, and will work with techniques and tools used in software development companies.

The course will be based on the development of a software project in Java in groups of 3-4 students. To do so, multiple projects will be offered and the groups will be able to chose which project they are interested in developing.

The students will put in practice the concepts explained in the theoretical sessions throughout the project development, such as how to face the design of the software, how to test their code, prepare the configuration scripts, document their code or manage the working in group issues.

## 5.2. Syllabus

1. Introducción to Software Development
2. Software Development Tools
  - 2.1. Version Control Systems
  - 2.2. Build Tools
3. Software Quality
  - 3.1. Technical Debt
  - 3.2. Static Analysis Tools
4. Software Testing
  - 4.1. Test automation frameworks
  - 4.2. Test-Driven Development
5. Software Development
  - 5.1. Continuous Integration
  - 5.2. Agile methodologies
6. Software design

## 6. Schedule

### 6.1. Subject schedule\*

Week	Face-to-face classroom activities	Face-to-face laboratory activities	Distant / On-line	Assessment activities
1	<b>Introduction to Software development</b> Duration: 02:00 Lecture		<b>Introduction to Software development</b> Duration: 02:00 Lecture	
2	<b>Version Control Systems</b> Duration: 02:00 Lecture		<b>Version Control Systems</b> Duration: 02:00 Lecture	
3	<b>Build tools</b> Duration: 02:00 Lecture		<b>Build tools</b> Duration: 02:00 Lecture	
4	<b>Software testing</b> Duration: 02:00 Lecture		<b>Software testing</b> Duration: 02:00 Lecture	
5	<b>Test automation frameworks</b> Duration: 02:00 Lecture		<b>Test automation frameworks</b> Duration: 02:00 Lecture	
6	<b>Test driven development</b> Duration: 02:00 Lecture		<b>Test driven development</b> Duration: 02:00 Lecture	
7				
8		<b>Laboratory Practice</b> Duration: 02:00 Laboratory assignments	<b>Laboratory Practice</b> Duration: 02:00 Laboratory assignments	<b>Laboratory practice</b> Group work Continuous assessment Not Presential Duration: 02:00
9		<b>Laboratory Practice</b> Duration: 02:00 Laboratory assignments	<b>Laboratory Practice</b> Duration: 02:00 Laboratory assignments	
10	<b>Software Development: Continuous Integration</b> Duration: 02:00 Lecture		<b>Software Development: Continuous Integration</b> Duration: 02:00 Lecture	
11	<b>Software Development: Agile Methodologies</b> Duration: 02:00 Lecture		<b>Software Development: Agile Methodologies</b> Duration: 02:00 Lecture	
12	<b>Software Design</b> Duration: 02:00 Lecture		<b>Software Design</b> Duration: 02:00 Lecture	
13	<b>Software Design</b> Duration: 02:00 Lecture		<b>Software Design</b> Duration: 02:00 Lecture	

14	<b>Project tutorial</b> Duration: 02:00 Laboratory assignments		<b>Project tutorial</b> Duration: 02:00 Laboratory assignments	
15	<b>Project tutorial</b> Duration: 02:00 Laboratory assignments		<b>Project tutorial</b> Duration: 02:00 Laboratory assignments	
16				<b>Written Exam</b> Written test Continuous assessment Presential Duration: 02:00  <b>Programming Project</b> Group work Continuous assessment and final examination Not Presential Duration: 20:00  <b>Written Exam</b> Written test Final examination Presential Duration: 02:00
17				

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
8	Laboratory practice	Group work	No Presential	02:00	10%	0 / 10	CG-6 CG-5 CG-19 CG-7:10/16/17 CG-2/CE45 CG-1/21 CG-3/4
16	Written Exam	Written test	Face-to-face	02:00	20%	4 / 10	CG-6 CG-5 CG-19 CG-1/21 CG-3/4
16	Programming Project	Group work	No Presential	20:00	70%	0 / 10	CG-1/21 CG-3/4 CG-7:10/16/17 CG-2/CE45 CG-6 CG-5 Ce 14/15 CG-19 CG-24/25/26/27

#### 7.1.2. Final examination

Week	Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
16	Programming Project	Group work	No Presential	20:00	70%	0 / 10	CG-1/21 CG-3/4 CG-7:10/16/17 CG-2/CE45 CG-6 CG-5 Ce 14/15 CG-19 CG-24/25/26/27

16	Written Exam	Written test	Face-to-face	02:00	30%	4 / 10	Ce 14/15 CG-6 CG-5 CG-19 CG-24/25/26/27 CG-7:10/16/17 CG-2/CE45 CG-1/21 CG-3/4
----	--------------	--------------	--------------	-------	-----	--------	--

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

Description	Modality	Type	Duration	Weight	Minimum grade	Evaluated skills
Written exam	Written test	Face-to-face	02:00	20%	4 / 10	CG-6 CG-5 Ce 14/15 CG-19 CG-24/25/26/27 CG-7:10/16/17 CG-2/CE45 CG-1/21 CG-3/4
Programming Project	Group work	Face-to-face	20:00	70%	0 / 10	CG-6 CG-5 Ce 14/15 CG-19 CG-24/25/26/27 CG-7:10/16/17 CG-2/CE45 CG-1/21 CG-3/4
Laboratory practice	Group work	Face-to-face	02:00	10%	0 / 10	CG-1/21 CG-2/CE45 CG-3/4 CG-5 CG-6 CG-19

## 7.2. Assessment criteria

### Continuous evaluation:

The course have two assignments that must be

- A laboratory practice, which will be done in pairs of two students around weeks 7-8.
- A programming project, which will be done in groups of 3-4 students and must be submitted before the date of the final exam. The project will have two submission dates, one intermediate deadline (3-4 weeks before the final deadline) and the final deadline when the project must be finished. The intermediate deadline could add an extra point in the grade of the Project.

The final grade of the course is divided in three parts:

- Laboratory practice (10%)
- Programming Project (70%)
- Written exam (20%)

### Final exam:

Those students who are interested in "final exam evaluation", according to the rules described in <https://www.fi.upm.es/?pagina=1147&idioma=english>, must notify to the coordinator no later than 15 calendar days after the beginning of the subject lectures.

The final grade for these students will have two parts:

- Programming Project (70%)
- Written exam (30%)

### Referred (re-sit) examination

The final grade for these students that did not pass the course will have three parts:

- Laboratory practice (10%)
- Programming Project (70%)
- Written exam (20%)

The grade obtained on January in the laboratory practice, the programming project and in the written exam will be saved for the extraordinary examination. Students can decide which parts of the course they want to repeat/improve on July.

## 8. Teaching resources

---

### 8.1. Teaching resources for the subject

Name	Type	Notes
Moodle	Web resource	All material will be available in online
Gitlab server	Equipment	A Gitlab server for working on the practical part of the course
SonarQube server	Equipment	A SonarQube server for the practical parts of the course