



UNIVERSIDAD
POLITÉCNICA
DE MADRID

PROCESO DE
COORDINACIÓN DE LAS
ENSEÑANZAS PR/CL/001



E.T.S. de Ingenieros
Informáticos

ANX-PR/CL/001-01

GUÍA DE APRENDIZAJE

ASIGNATURA

105000394 - Programming Project

PLAN DE ESTUDIOS

10II - Grado En Ingenieria Informatica

CURSO ACADÉMICO Y SEMESTRE

2025/26 - Segundo 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.....	3
6. Cronograma.....	5
7. Actividades y criterios de evaluación.....	7
8. Recursos didácticos.....	10
9. Otra información.....	11

1. Datos descriptivos

1.1. Datos de la asignatura

Nombre de la asignatura	105000394 - Programming Project
No de créditos	3 ECTS
Carácter	Optativa
Curso	Tercero curso
Semestre	Sexto semestre
Período de impartición	Febrero-Junio
Idioma de impartición	Inglés/Castellano
Titulación	10II - Grado en Ingeniería Informática
Centro responsable de la titulación	10 - E.T.S. De Ingenieros Informáticos
Curso académico	2025-26

2. Profesorado

2.1. Profesorado implicado en la docencia

Nombre	Despacho	Correo electrónico	Horario de tutorías *
Guillermo Roman Diez (Coordinador/a)	2304	guillermo.roman@upm.es	L - 12:00 - 15:00 X - 12:00 - 15:00 Please send an e-mail to set up an appointment
Raul Alonso Calvo	2315	raul.alonso@upm.es	L - 10:00 - 13:00 X - 10:00 - 13:00 Please send an e-mail to set up an appointment

* 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 Arte De Programar
- Programación II
- Algoritmos Y Estructura De Datos
- Programación I
- Programacion Para Sistemas
- Concurrencia

3.2. Otros conocimientos previos recomendados para cursar la asignatura

- Programación de interfaces de usuario

4. Competencias y resultados de aprendizaje

4.1. Competencias

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. Resultados del aprendizaje

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

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

5. Descripción de la asignatura y temario

5.1. Descripción de la asignatura

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. Temario de la asignatura

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

6. Cronograma

6.1. Cronograma de la asignatura *

Sem	Actividad tipo 1	Actividad tipo 2	Tele-enseñanza	Actividades de evaluación
1	Introduction to Software development Duración: 02:00 LM: Actividad del tipo Lección Magistral			
2	Version Control Systems Duración: 02:00 LM: Actividad del tipo Lección Magistral			
3	Build tools Duración: 02:00 LM: Actividad del tipo Lección Magistral			
4	Software testing Duración: 02:00 LM: Actividad del tipo Lección Magistral			
5	Test automation frameworks Duración: 02:00 LM: Actividad del tipo Lección Magistral			
6	Test driven development Duración: 02:00 LM: Actividad del tipo Lección Magistral			
7	Laboratory Practice Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio			
8	Laboratory Practice Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio			Laboratory practice (NON RECOVERABLE) TG: Técnica del tipo Trabajo en Grupo Evaluación Progresiva No presencial Duración: 02:00
9	Software Development: Continuous Integration Duración: 02:00 LM: Actividad del tipo Lección Magistral			
10	Software Development: Agile Methodologies Duración: 02:00 LM: Actividad del tipo Lección Magistral			
11	Software Design Duración: 02:00 LM: Actividad del tipo Lección Magistral			
12	Software Design Duración: 02:00 LM: Actividad del tipo Lección Magistral			

13	Project tutorial Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio			
14	Project tutorial Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio			
15				
16	Written exam Duración: 01:00 OT: Otras actividades formativas / Evaluación			Written Exam EX: Técnica del tipo Examen Escrito Evaluación Progresiva y Global Presencial Duración: 01:00 Programming Project TG: Técnica del tipo Trabajo en Grupo Evaluación Progresiva y Global No presencial Duración: 20:00
17				

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
8	Laboratory practice (NON RECOVERABLE)	TG: Técnica del tipo Trabajo en Grupo	No Presencial	02:00	10%	0 / 10	CG-1/21 CG-2/CE45 CG-3/4 CG-5 CG-6 CG-7:10/16/17 CG-19
16	Written Exam	EX: Técnica del tipo Examen Escrito	Presencial	01:00	20%	4 / 10	CG-1/21 CG-3/4 CG-5 CG-6 CG-19
16	Programming Project	TG: Técnica del tipo Trabajo en Grupo	No Presencial	20:00	70%	5 / 10	CG-1/21 CG-2/CE45 CG-3/4 CG-5 CG-6 CG-7:10/16/17 CG-19 CG-24/25/26/27 Ce 14/15

7.1.2. Prueba evaluación global

Sem	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
16	Written Exam	EX: Técnica del tipo Examen Escrito	Presencial	01:00	20%	4 / 10	CG-1/21 CG-3/4 CG-5 CG-6 CG-19
16	Programming Project	TG: Técnica del tipo Trabajo en Grupo	No Presencial	20:00	70%	5 / 10	CG-1/21 CG-2/CE45 CG-3/4 CG-5 CG-6 CG-7:10/16/17 CG-19 CG-24/25/26/27 Ce 14/15

7.1.3. Evaluación convocatoria extraordinaria

Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
Written exam	EX: Técnica del tipo Examen Escrito	Presencial	02:00	20%	4 / 10	CG-1/21 CG-2/CE45 CG-3/4 CG-5 CG-6 CG-7:10/16/17 CG-19 CG-24/25/26/27 Ce 14/15
Programming Project	TG: Técnica del tipo Trabajo en Grupo	Presencial	20:00	70%	5 / 10	CG-1/21 CG-2/CE45 CG-3/4 CG-5 CG-6 CG-7:10/16/17 CG-19 CG-24/25/26/27 Ce 14/15

7.2. Criterios de evaluación

Progressive evaluation:

The course have two practical assignments that must be done along the course:

- 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 (4 weeks before the final deadline) and the final deadline when the project must be completely finished.

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

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

Laboratory practice is non recoverable as its main goal the environment preparation and to be familiar with the tools that will be used in the project. For these reasons, it does not make sense to do the laboratory practice in the referred evaluation.

As the written exam is a multiple choice exam, for academic reasons, the solution of the exam will not be publish.

Global evaluation:

The global evaluation includes the same evaluation parts that the progressive evaluation. As the written exam and the project are done in the official evaluation period, there is no option to recover them in the global evaluation. Thus, the final grade for these students will have the same three parts than before:

- Laboratory practice (10%) - cannot be submitted again as it is non recoverable
- Programming Project (70%)
- Written exam (20%)

Referred (re-sit) examination

In the referred evaluation, students can submit their projects and have another chance to pass the written exam. The grades obtained in the laboratory practice, the programming project and in the written exam will be saved for the referred examination. Students can decide which parts of the course they would like to repeat on the referred evaluation. The final grade for the referred evaluation will have three parts:

- Laboratory practice (10%) - cannot be submitted again as it is non recoverable
- Programming Project (70%)
- Written exam (20%)

IMPORTANT: A system for detecting plagiarisms will be used and the groups involved in plagiarism will be penalized with a failing grade in the corresponding examination period according to the UPM norms.

8. Recursos didácticos

8.1. Recursos didácticos de la asignatura

Nombre	Tipo	Observaciones
Moodle	Recursos web	All material will be available in online
Gitlab server	Equipamiento	A Gitlab server for working on the practical part of the course
SonarQube server	Equipamiento	A SonarQube server for the practical parts of the course
Practical TDD and Acceptance TDD for Java Developers	Bibliografía	L. Koskela, Test Driven: Practical TDD and Acceptance TDD for Java Developers, Manning Publications Co., Greenwich, CT, USA, 2007.
Clean code: A handbook of agile software craftsmanship	Bibliografía	Robert C. Martin, Clean code: A handbook of agile software craftsmanship, 1 ed., Prentice Hall PTR, Upper Saddle River, NJ, USA, 2008.
The art of agile development	Bibliografía	James Shore and Shane Warden, The art of agile development, first ed., O'Reilly, 2007

9. Otra información

9.1. Otra información sobre la asignatura

There is no extra information regarding the course.

Metodologías docentes innovadoras aplicadas en la asignatura:

Learning by doing: Los alumnos deben enfrentarse a lo largo de la asignatura al desarrollo de dos proyectos software, uno de ellos en parejas y el otro en grupos de 3 o 4. Cada uno de estos retos prácticos tienen una temática realista y para cada una de las prácticas los alumnos reciben una serie de instrucciones. Durante sesiones prácticas (6 a lo largo del semestre), el profesor hace una breve descripción del problema a resolver, incluyendo algunos consejos para afrontarlo, y los estudiantes disponen de tiempo con el profesor para resolver las posibles dudas que les puedan surgir.

Aprendizaje basado en retos: La metodología Learning by Doing se integra a la perfección con el aprendizaje basado en retos. Cada uno de los proyectos propuestos es un reto, dado que su duración es de 2 y 8 semanas respectivamente, permitiendo a los alumnos aplicar los conocimientos adquiridos a lo largo de los años de carrera.