



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

105000016 - Programacion Para Sistemas

PLAN DE ESTUDIOS

10II - Grado En Ingenieria Informatica

CURSO ACADÉMICO Y SEMESTRE

2025/26 - Primer semestre

Índice

Guía de Aprendizaje

1. Datos descriptivos.....	1
2. Profesorado.....	1
3. Conocimientos previos recomendados.....	3
4. Competencias y resultados de aprendizaje.....	3
5. Descripción de la asignatura y temario.....	4
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	105000016 - Programacion para Sistemas
No de créditos	3 ECTS
Carácter	Obligatoria
Curso	Segundo curso
Semestre	Tercer semestre
Período de impartición	Septiembre-Enero
Idioma de impartición	Castellano
Titulación	10II - Grado en Ingenieria Informatica
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 *
Santiago Tapia Fernandez		santiago.tapia@upm.es	Sin horario. Consultar las tutorías en el Aula Virtual de la asignatura. Más información en http://dlsiis.fi.upm.es/tutorias-2526

Angel Herranz Nieva		angel.herranz@upm.es	Sin horario. Consultar las tutorías en el Aula Virtual de la asignatura. Más información en http://dlsiis.fi.upm.es/tutorias-2526
Jose Crespo Del Arco		jose.crespo@upm.es	Sin horario. Consultar las tutorías en el Aula Virtual de la asignatura. Más información en http://dlsiis.fi.upm.es/tutorias-2526
Julio Setien Villaran	5207	julio.setien@upm.es	Sin horario. Consultar las tutorías en el Aula Virtual de la asignatura. Más información en http://dlsiis.fi.upm.es/tutorias-2526
Juan Luis Perez Camaño (Coordinador/a)	L5002	juanluis.perez@upm.es	Sin horario. Consultar las tutorías en el Aula Virtual de la asignatura. Más información en http://dlsiis.fi.upm.es/tutorias-2526
Santiago Eibe Garcia		santiago.eibe@upm.es	Sin horario. Consultar las tutorías en el Aula Virtual de la asignatura. Más

			información en http://dlsiis.fi.upm.es/tutorias-2526
--	--	--	--

* 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

- Programación I

3.2. Otros conocimientos previos recomendados para cursar la asignatura

- Capacidad de comunicación oral y escrita en español

- Desarrollar programas sencillos en un lenguaje de programación de propósito general.

4. Competencias y resultados de aprendizaje

4.1. Competencias

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

Ce 8 - Poseer destrezas fundamentales de la programación que permitan la implementación de los algoritmos y las estructuras de datos en el software.

Ce 9 - Poseer las destrezas que se requieren para diseñar e implementar unidades estructurales mayores que utilizan los algoritmos y las estructuras de datos, así como las interfaces por las que se comunican estas unidades.

4.2. Resultados del aprendizaje

RA431 - Programar aplicaciones, en al menos un lenguaje procedimental ampliamente utilizado, como puede ser C, que le permiten la comunicación con el sistema

RA430 - Programar scripts que automaticen determinadas tareas o faciliten llevar a cabo pruebas funcionales de programas

RA291 - Programar aplicaciones, en al menos un lenguaje procedimental ampliamente utilizado, como puede ser C, que le permiten la comunicación con el sistema.

RA292 - Programar scripts que automaticen determinadas tareas o faciliten llevar a cabo pruebas funcionales de programas.

5. Descripción de la asignatura y temario

5.1. Descripción de la asignatura

El principal objetivo de la asignatura es que los alumnos adquieran conocimientos prácticos para al desarrollo de programas informáticos en el lenguaje C y programación de scripts en el lenguaje Shell. Aunque la asignatura tiene una parte específica para programación en C (11 semanas) y otra de lenguaje Shell (4 semanas) ambas partes están totalmente relacionadas e integradas en la actividad docente de la asignatura.

5.2. Temario de la asignatura

1. Sistema operativo Unix y shell Bash

1.1. Introducción a Unix y al intérprete de mandatos. Lenguaje de Shell. Comandos y programas útiles.

1.2. Programación con scripts. Automatización de tareas.

2. Lenguaje de programación C y herramientas de desarrollo.

2.1. Herramientas de desarrollo y entorno UNIX: editor, compilador, enlazador, depurador.

2.2. Programación estructurada y fundamentos de C. Tipos de datos, estructuras de control, Entrada / Salida, funciones

2.3. Programación C avanzada. Arrays, strings, punteros y memoria dinámica

6. Cronograma

6.1. Cronograma de la asignatura *

Sem	Actividad tipo 1	Actividad tipo 2	Tele-enseñanza	Actividades de evaluación
1	Tema 1.1: clase de teoría Duración: 02:00 LM: Actividad del tipo Lección Magistral			Actividades en el aula. Durante el curso en horario de clase se realizarán pruebas prácticas y teóricas. OT: Otras técnicas evaluativas Evaluación Progresiva Presencial Duración: 00:10
2	Tema 1.2: clase de teoría Duración: 02:00 LM: Actividad del tipo Lección Magistral			
3	Tema 1,1, Ejercicios Unix Duración: 02:00 PR: Actividad del tipo Clase de Problemas			
4	Tema 1.2, Ejercicios Scripts Duración: 02:00 PR: Actividad del tipo Clase de Problemas			
5	Tema 2.1 : Clase de teoría. Duración: 02:00 LM: Actividad del tipo Lección Magistral			
6	Tema 2.2: clase teoría Duración: 02:00 LM: Actividad del tipo Lección Magistral			
7	Tema 2.1, ejercicios prácticos C Duración: 02:00 PR: Actividad del tipo Clase de Problemas			
8	Tema 2.2, ejercicios prácticos C Duración: 02:00 PR: Actividad del tipo Clase de Problemas			
9	Tema 2.2, ejercicios prácticos C Duración: 02:00 PR: Actividad del tipo Clase de Problemas			
10	Tema 2. 3: clase teoría Duración: 02:00 LM: Actividad del tipo Lección Magistral			Práctica entrega 1 TI: Técnica del tipo Trabajo Individual Evaluación Progresiva No presencial Duración: 00:10
11	Tema 2.3, ejercicios prácticos C Duración: 02:00 PR: Actividad del tipo Clase de Problemas			
12	Tema 2.3, ejercicios prácticos C Duración: 02:00 PR: Actividad del tipo Clase de Problemas Examen parcial práctico Duración: 02:00 OT: Otras actividades formativas / Evaluación			Examen Parcial práctico. EP: Técnica del tipo Examen de Prácticas Evaluación Progresiva Presencial Duración: 02:00

13	Tema 2.3: clase teoría Duración: 02:00 LM: Actividad del tipo Lección Magistral			
14	Tema 2.3, ejercicios prácticos C Duración: 02:00 PR: Actividad del tipo Clase de Problemas			
15	Tema 2.3, ejercicios prácticos C Duración: 02:00 PR: Actividad del tipo Clase de Problemas			Práctica entrega 2 TI: Técnica del tipo Trabajo Individual Evaluación Progresiva No presencial Duración: 00:30
16				
17	Examen teórico Duración: 00:30 OT: Otras actividades formativas / Evaluación Examen práctico Duración: 02:00 OT: Otras actividades formativas / Evaluación			Examen teórico ET: Técnica del tipo Prueba Telemática Evaluación Progresiva Presencial Duración: 00:30 Examen práctico EP: Técnica del tipo Examen de Prácticas Evaluación Progresiva Presencial Duración: 02:00 Examen práctico. EP: Técnica del tipo Examen de Prácticas Evaluación Global Presencial Duración: 02:00 Examen teórico ET: Técnica del tipo Prueba Telemática Evaluación Global Presencial Duración: 00:30

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
1	Actividades en el aula. Durante el curso en horario de clase se realizarán pruebas prácticas y teóricas.	OT: Otras técnicas evaluativas	Presencial	00:10	10%	0 / 10	CG-19 Ce 8 Ce 9
10	Práctica entrega 1	TI: Técnica del tipo Trabajo Individual	No Presencial	00:10	5%	0 / 10	CG-19 Ce 8 Ce 9
12	Examen Parcial práctico.	EP: Técnica del tipo Examen de Prácticas	Presencial	02:00	20%	0 / 10	CG-19 Ce 8 Ce 9
15	Práctica entrega 2	TI: Técnica del tipo Trabajo Individual	No Presencial	00:30	5%	0 / 10	Ce 9 CG-19 Ce 8
17	Examen teórico	ET: Técnica del tipo Prueba Telemática	Presencial	00:30	30%	4 / 10	CG-19 Ce 8 Ce 9
17	Examen práctico	EP: Técnica del tipo Examen de Prácticas	Presencial	02:00	30%	4 / 10	CG-19 Ce 8 Ce 9

7.1.2. Prueba evaluación global

Sem	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
17	Examen práctico.	EP: Técnica del tipo Examen de Prácticas	Presencial	02:00	70%	5 / 10	CG-19 Ce 8 Ce 9

17	Examen teórico	ET: Técnica del tipo Prueba Telemática	Presencial	00:30	30%	5 / 10	CG-19 Ce 8 Ce 9
----	----------------	--	------------	-------	-----	--------	-----------------------

7.1.3. Evaluación convocatoria extraordinaria

Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
Examen teórico	ET: Técnica del tipo Prueba Telemática	Presencial	00:30	30%	5 / 10	Ce 9 CG-19 Ce 8
Examen práctico	EP: Técnica del tipo Examen de Prácticas	Presencial	02:30	70%	5 / 10	Ce 9 CG-19 Ce 8

7.2. Criterios de evaluación

EVALUACIÓN PROGRESIVA:

- **Actividades de aula (10 %):** Los alumnos realizarán actividades evaluables de tipo teórico y práctico en el aula. Solo se tendrán en cuenta el 75% (mejor) de las actividades pedidas por si alguna alumno no ha podido asistir a alguna clase.
- **Examen parcial (20%) :** Los alumnos resolverán durante el examen ejercicios de C y Unix del temario explicado hasta la fecha.
- **Práctica entrega 1 (5%) y entrega 2 (5%) :** Los alumnos deberán entregar los ejercicios solicitados.
- **Examen teórico (30 %) :** Examen on-line con preguntas cortas y/o tipo test sobre C y Bash. **Nota mínima 4.**
- **Examen práctico (30%):** Los alumnos tendrán que implementar en C y Bash los ejercicios pedidos. **Nota mínima 4.**

Para ser evaluado por progresiva se requiere que $(0.25 \times \text{Actividades de aula}) + (0.5 \times \text{Examen parcial}) + (0.125 \times \text{Práctica entrega 1}) + (0.125 \times \text{Práctica entrega 2}) \geq 5$. Es decir se requiere al menos 2 de los 4 puntos que se otorgan durante el curso para ser evaluado por progresiva.

NOTA_PROGRESIVA = (0.1xActividades de aula) + (0.2xExamen parcial) + (0.05xPráctica entrega 1) + (0.05xPráctica entrega 2) + (0.3 x Examen teórico) + (0.3 x Examen práctico).

NOTA_PROGRESIVA_FINAL = MAX (NOTA_PROGRESIVA, (0.3 x Examen teórico+ 0.7 x Examen práctico))

EVALUACIÓN GLOBAL:

Aquellos alumnos que no hayan alcanzado a obtener los 2 puntos de los 4 que se otorgan durante el curso como se explica en el apartado de progresiva serán evaluado por evaluación global:

- Examen teórico (30 %) : Examen on-line con preguntas cortas y/o tipo test sobre C y Bash. Nota mínima 5.
- Examen práctico (70%): Los alumnos tendrán que implementar en C y Bash los ejercicios pedidos. Nota mínima 5.

NOTA_GLOBAL_FINAL = (0.3 x Examen teórico)+ (0.7 x Examen práctico)

CONVOCATORIA EXTRAORDINARIA:

Aquellos alumnos que no hayan aprobado la asignatura en convocatoria ordinaria se examinarán en la convocatoria extraordinaria.

- Examen teórico (30 %) : Examen on-line con preguntas cortas y/o tipo test sobre C y Bash. Nota mínima 5.
- Examen práctico (70%): Los alumnos tendrán que implementar en C y Bash los ejercicios pedidos. Nota mínima 5.

NOTA_EXTRAORDINARIA_FINAL = (0.3 x Examen teórico)+ (0.7 x Examen práctico)

CONSIDERACIONES GENERALES PARA LA EVALUACIÓN

- Para las clases, las prácticas y los exámenes los alumnos deben disponer de un ordenador y un entorno

local Linux (o a través de Escritorios Virtuales) donde puedan realizar los ejercicios pedidos. Además deberá poder trasladar los códigos a la máquina triqui para su compilación y entrega.

- Todas las entregas de código (exámenes y/o prácticas) deberán compilar en un entorno Linux.

- No se guardan notas entre cursos, ni entre convocatorias, ni entre evaluaciones.

- Todas las pruebas serán anunciadas vía Moodle a los alumnos. Si por alguna necesidad justificada (aulas disponibles, coincidencia con otras pruebas, ...) fuera necesario modificar ligeramente la semana de realización de alguna de las pruebas se comunicará inmediatamente a los alumnos la nueva fecha.

- Si en alguna convocatoria no se supera alguna de las notas mínimas requeridas, la calificación en esa convocatoria será de 3 "Suspenso".

- La utilización de aplicaciones/software no permitido durante un examen/práctica supondrá ser calificado como Fraude Académico.

8. Recursos didácticos

8.1. Recursos didácticos de la asignatura

Nombre	Tipo	Observaciones
Material de apoyo	Otros	Material didáctico y de apoyo proporcionado por los profesores de la asignatura.
Antonakos	Bibliografía	Programación estructurada en C, J.L. Antonakos, K.C. Mansfield Jr. Prentice-Hall 1997.
Hoover	Bibliografía	System Programming with C and Unix, A. Hoover, Pearson Education, 2009
Kernighan-Pike	Bibliografía	The UNIX programming environment, B. Kernighan, R. Pike, Second Edition. Prentice-Hall 1988.

Kerninghan-Ritchie	Bibliografía	The C programming language, B. Kerninghan, D.Ritchie. Segunda edición. Prentice-Hall 1988.
Newham	Bibliografía	Learning the bash shell, C. Newham, B. Rosenblatt. O'Reilly 2005.
Manual Bash	Bibliografía	The GNU Bash Reference Manual (revised for version 3.2), Chet Ramey and Brian Fox http://www.network-theory.co.uk/bash/manual
Aula virtual	Recursos web	Página web de la asignatura: http://web3.fi.upm.es/AulaVirtual/course/view.php?id=135
Hardware	Equipamiento	Laboratorio: salas de ordenadores
Software	Equipamiento	Software: sistema operativo GNU/Linux + herramientas de desarrollo C y Bash (gcc, gdb/ddd, make, bash).

9. Otra información

9.1. Otra información sobre la asignatura

Los alumnos realizarán ejercicios en clase y exámenes con sus propios portátiles, donde tendrán que tener instalado todo el entorno necesario para su realización. Si algún alumno no tuviera portátil, la Biblioteca del Centro dispone de un servicio de préstamo.

El cronograma sigue una planificación teórica de la asignatura y puede sufrir modificaciones durante el curso. Esta Guía de Aprendizaje es la referencia general para esta asignatura. La información real y actualizada sobre su implementación en el semestre corriente (calendario, horario, fechas, plazos, pesos, avisos, listas, etc.), se publicará con antelación en el curso Moodle de la asignatura. Cualquier conflicto, deficiencia, inconsistencia o discrepancia entre la información de esta guía y la publicada en el curso Moodle deberá ser resuelta en favor de este segundo.

Metodologías docentes innovadoras aplicadas en la asignatura

En la asignatura se implementan varias metodologías docentes innovadoras

(<https://innovacioneducativa.upm.es/guias-pdi>) con el fin de motivar y reforzar el aprendizaje por parte del

estudiantado:

Learning by Doing: A los alumnos se les propone una serie de ejercicios prácticos de programación en C que tienen que resolver y en el proceso van descubriendo los errores que cometen, sobre todo en el proceso de compilación. Los alumnos son autónomos en el aprendizaje con la supervisión del profesor.

Gamificación: A través de la realización de ejercicios y actividades aula (con herramientas como Wooclap o similares), se han dinamizado las clases con la participación de los estudiantes presentes en el aula, resolviendo preguntas en tiempo real que permite tener un feedback inmediato. Los resultados obtenidos por los alumnos formaban parte de la evaluación de la asignatura.