

Plan docente de Programación II

1. Descripción y contextualización

<i>Identificación y características de la materia</i>				
<i>Denominación y código</i>	Programación II			
<i>Curso y Titulación</i>	2º Ing. Técnica en Telecomunicación, Especialidad Telemática (225 LRU)			
<i>Área</i>	Lenguajes y Sistemas Informáticos			
<i>Departamento</i>	<i>Informática</i>			
<i>Tipo</i>	TR			
<i>Coefficientes</i>	Practicidad: 4		Agrupamiento: 2 (Medio)	
<i>Duración ECTS (créditos)</i>	6 ECTS (150 horas)			
<i>Distribución ECTS (rangos)</i>	Grupo Grande:	Seminario-Lab.:	Tutoría ECTS:	No presenciales:
	24,67% 37	19,33% 29	2% 3	54% 81
<i>Descriptores (según BOE)</i>	1C			
<i>Coordinador-Profesor/es</i>	Luis Jesús Arévalo Rosado			

2. Objetivos

<i>RELACIONADOS CON COMPETENCIAS ACADÉMICAS Y DISCIPLINARES</i>	<i>VINCULACIÓN</i>
Descripción	CET ⁱ
1. Afianzar al alumno en el desarrollo de programas informáticos basados en la metodología de programación estructurada.	4,6
2. Iniciar al alumno a la programación orientación a objetos	4,6
3. Permitir a los alumnos resolver supuestos que supongan la implementación de programas de tamaño medio, con codificaciones correctas y eficientes, utilizando el paradigma de la programación estructurada junto con la orientada a objetos.	4,6
4. Preparar al alumno para la implementación de programas de gran tamaño y complejidad.	4,6
5. Presentar las estructuras de datos lineales básicas y orientar al alumno para que sea capaz de seleccionar la más adecuada en cada problema.	4,6
6. Presentar las estructuras de datos no lineales básicas y orientar al alumno para cuando debe usarse.	4,6
7. Capacitar al alumno para el tratamiento de información almacenada en dispositivos de almacenamiento secundario	4,6
<i>RELACIONADOS CON OTRAS COMPETENCIAS PERSONALES Y PROFESIONALES</i>	<i>VINCULACIÓN</i>
Descripción	CET
1. Adquirir la capacidad de analizar, diseñar e implementar en un desarrollo software las necesidades para poder resolver un problema informático.	1,2,5,7,13,15
2. El alumno debe conocer los principales conceptos de un lenguaje de programación orientado a objetos.	1,2,5,7,13,15
3. Desarrollar la capacidad de dimensionar y relativizar los problemas, aplicando resultados menores en la consecución de proyectos más ambiciosos.	1,2,5,7
4. Trabajar en equipo	9,12

3. Contenidos

Secuenciación de bloques temáticos y temas

TEMA 1. Punteros y referencias

1. Introducción
2. Apuntadores
3. Fundamentos
4. Operadores
 5. Operaciones Básicas
 6. Otras características
 7. Punteros y arrays
 8. Punteros y estructuras
9. Gestión dinámica de memoria
10. Referencias
11. Bibliografía

TEMA 2. Programación orientada a objetos

1. Introducción
2. Declaración de una clase
3. Encapsulación
4. Representación gráfica de una clase
5. Constructores y destructores
6. Creación y eliminación dinámica de objetos
7. Paso y retorno de objetos a funciones
8. Otras características
 1. Operador asignación
 2. Operador this
 3. Atributos estáticos
 4. Métodos constantes
9. Bibliografía
10. Ejemplos y ejercicios

TEMA 3. Sobrecarga de operadores

1. Introducción. Concepto de sobrecarga.
2. Sobrecarga de funciones.
3. Sobrecarga de métodos.
4. Sobrecarga de operadores.
 1. Sobrecarga de operadores unarios.
 2. Sobrecarga de operadores binarios.
 3. Sobrecarga como métodos y como funciones amigas.
5. Sobrecarga del operador <<
6. Ejemplos.

TEMA 4. Herencia y polimorfismo

1. Introducción a la Herencia
 1. Tipos de herencia
 1. Herencia simple
 2. Herencia múltiple
 3. Herencia privada
 4. Herencia pública
 5. Herencia protegida
 2. Constructores en herencia
 3. Destructores en herencia
 4. Métodos: Sobrecarga vs Redefinición

5. Asignación de instancias
6. Ejercicios y problemas
2. Introducción al polimorfismo
 1. Punteros a clases derivadas
 2. Funciones virtuales
 3. Funciones virtuales puras
 4. Clases abstractas
 5. Vinculación dinámica
 6. Ejercicios y problemas

TEMA 5. Excepciones

- Introducción
- Estructura de excepciones
- Excepciones definidas por el usuario
- Clase “exception”
- Ejercicios y problemas

TEMA 6. Estructura de datos lineales

- Introducción
- TAD Pila
 - Descripción
 - Especificación algebraica
 - Especificación de la clase Pila
 - Implementaciones de la clase Pila
- TAD Cola
 - Descripción
 - Especificación algebraica
 - Especificación de la clase Cola
 - Implementaciones de la clase Cola
- TAD Lista
 - Descripción
 - Especificación algebraica
 - Especificación de la clase
 - Implementaciones de la clase Lista
- Ejercicios y problemas

TEMA 7. Estructuras de Datos Avanzadas: Árboles

- Introducción
- TAD Arbol
- TAD Arbol Ordenado
- Otros conceptos
- TAD Árbol binario
- Ejercicios y problemas

TEMA 8. Ficheros

1. Introducción
2. Operaciones básicas con ficheros
3. Apertura y cierre
4. Lectura y Escritura
5. Acceso aleatorio
6. Errores de entrada / salida
7. Ejercicios y problemas

<i>Interrelación</i>				
Requisitos (Rq) y redundancias (Rd)		Tema	Procedencia	
Programación imperativa		Rq	1-11	Programación I, Transmisión de datos, Fundamentos de telemática
Vectores y Punteros		Rq	8,9, 11	Programación I
Conocimientos matemáticos básicos		Rq	1-9	Análisis Matemático I

4. Metodología docente y plan de trabajo del estudiante

<i>Actividades de enseñanza-aprendizaje</i>				<i>Vinculación</i>	
<i>Descripción y secuenciación de actividades</i>	<i>Tipoⁱⁱ</i>		<i>Dⁱⁱⁱ</i>	<i>Tema</i>	<i>Objet.</i>
1. Presentación de la asignatura	CE	I	0,5	1-7	
2. Encuesta de conocimientos previos	CE	I	0,5	1-7	
3. Práctica 0. Repaso de programación imperativa	S/L	P-V	3	-	1
4. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 1	NP	T-II	1	1	1
5. Exposición sobre conceptos de programación dinámica (Tema 1)	GG	T-III	2	1	1
6. Estudio de los contenidos explicados en el tema 1	NP	T-II	2	1	1
7. Práctica 1. Programación dinámica	S/L	P-V	3	1	2
8. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 2	NP	T-II	2	2	2
9. Exposición sobre conceptos de programación orientada a objetos (Tema 2)	GG	T-III	3	2	2
10. Estudio de los contenidos explicados en el tema 2	NP	T-II	3	2	2
11. Resolución de problemas de pizarra sobre POO (Tema 2)	GG	VII	2	2	2
12. Práctica 2. Programación orientada a objetos	S/L	P-V	3	2	2
13. Desarrollo y entrega de un caso práctico de POO	NP	P-V	3	2	2
14. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 3.	NP	T-II	2	3	2
15. Exposición sobre conceptos de sobrecarga. Tema 3.	GG	T-III	2	3	2
16. Estudio de los contenidos explicados en el tema 3	NP	T-II	2	3	2
17. Práctica 3. Sobrecarga y Composición	S/L	P-V	3	2,3	2
18. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 4.	NP	T-II	2	4	2
19. Exposición sobre conceptos de herencia y polimorfismo (Tema 4)	GG	T-III	2	4	2
20. Estudio de los contenidos explicados en el tema 4	NP	T-II	2	4	2
21. Resolución de problemas de pizarra sobre herencia y polimorfismo (tema 4)	GG	VII	2	4	2
22. Práctica 4. Herencia y polimorfismo	S/L	P-V	5	4	2
23. Desarrollo y entrega de un caso práctico sobre herencia y polimorfismo	NP	P-V	4	4	2
24. Tutorías ECTS	ECTS	I	1	1-5	Todos
25. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 5.	NP	T-II	1	5	2
26. Exposición sobre conceptos de excepciones	GG	T-III	2	5	2
27. Estudio de los contenidos explicados en el tema 5	NP	T-II	1	5	2
28. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 6.	NP	T-II	2	5	3
29. Exposición sobre conceptos de Estructuras de Datos Lineales	GG	T-III	4	6	3
30. Estudio de los contenidos explicados en el tema 6	NP	T-II	4	6	3
31. Resolución de problemas de pizarra sobre el Est. Datos Lineales	GG	VII	3	6	3
32. Práctica 5. Est. Datos Lineales	S/L	P-V	7	6	3

33. Desarrollo y entrega de un caso práctico sobre Est. Datos Lineales	NP	P-V	6	6	3
34. Tutorías ECTS	ECTS	I	1	6-7	Todos
35. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 7.	NP	T-II	2	7	4
36. Exposición sobre conceptos de Estructuras de Datos no Lineales	GG	T-III	4	7	4
37. Estudio de los contenidos explicados en el tema 7	NP	T-II	5	7	4
38. Resolución de problemas de pizarra sobre el Est. Datos no Lineales	GG	VII	3	7	4
39. Práctica 6. Est. Datos no Lineales	S/L	P-V	3	7	4
40. Desarrollo y entrega de un caso práctico sobre Est. Datos no Lineales	NP	P-V	6	7	4
41. Lectura previa de los resúmenes y desarrollos escritos de cada tema proporcionado antes de su impartición. Tema 8.	NP	T-II	2	8	4
42. Exposición sobre conceptos de ficheros	GG	T-III	2	8	4
43. Estudio de los contenidos explicados en el tema 8	NP	T-II	2	8	4
44. Práctica 7. Gestión de ficheros	S/L	P-V	2	8	4
45. Desarrollo y entrega de un caso práctico final	NP	P-V	9	1-8	Todos
46. Elaboración de memorias de prácticas de laboratorio	NP	VII	6	1-8	Todos
47. Tutorías ECTS	ECTS	I	1	1-8	Todos
48. Resolución de dudas	GG	T-III	2	1-8	Todos
49. Asistencia a tutorías complementarias para aclaración de dudas	NP	Tut	3	1-8	Todos
50. Preparación del examen final	NP	T-III	9	1-8	Todos
51. Examen Final	GG	VII	4	1-8	Todos

<i>Distribución de actividades</i>		<i>Nº alumnos</i>	<i>Dedicación del profesor</i>			
			<i>H. presenciales</i>	<i>H. no presenc.</i>	<i>H. presenciales</i>	<i>H. no presenc.</i>
Grupo grande (Más de 20 alumnos)	Coordinac./evaluac. (I)	40	1	-	1	5
	Teóricas (II y III)	40	22	35	22	22
	Prácticas (IV, V y VI)	40	14	9	14	10
	Subtotal	40	37	44	37	37
Seminario-Laboratorio (6-20 alumnos)	Coordinac./evaluac. (I)	20				
	Teóricas (II y III)	20				
	Prácticas (IV, V y VI)	20	29	28	58	29
	Subtotal	20	29	28	58	29
Tutoría ECTS (1-5 alumnos)	Coordinac./evaluac. (I)	10	3		12	10
	Teóricas (II y III)	10				
	Prácticas (IV, V y VI)	10				
	Subtotal	10	3		12	10
Tutoría comp. y preparación de ex. (VII)		1		9		12
Totales			69 (2,76 ECTS)	81 (3,24 ECTS)	107	88

Otras consideraciones metodológicas

Recursos y metodología de trabajo en las actividades presenciales

Las clases teóricas consistirán en la explicación del tema por parte del profesor y la realización de ejercicios prácticos sobre programación por parte del alumnado.

Recursos y metodología de trabajo en las actividades semi-presenciales y no presenciales

Una vez que el tema se ha explicado, los alumnos mediante una plataforma virtual realizarán cuestionarios sobre los aspectos desarrollado en las clases teóricas. Por otro lado mediante la entrega de prácticas se valorará al alumnado el conocimiento adquirido.

Recursos y metodología de trabajo para los alumnos que no han alcanzado los requisitos

Los alumnos deberán presentar en septiembre aquéllas partes e la asignatura no superada en febrero.

Recursos y metodología de trabajo para desarrollar competencias transversales

Para el desarrollo de competencias transversales, los alumnos utilizarán el trabajo en grupo durante las sesiones de resolución de ejercicios y el desarrollo de las prácticas que les ayuda a resolver un problema complejo.

5. Evaluación

<i>Criterios de evaluación* (1er cuatrimestre)</i>	<i>Vinculación*</i>	
<i>Descripción</i>	<i>Objetivo</i>	<i>CCⁿ</i>
1. Demostrar la adquisición y comprensión de los conceptos fundamentales de la asignatura (Test)	1,2,3,4,5	15%
2. Resolver problemas aplicando los conocimientos teóricos y los conocimientos adquiridos en los supuestos prácticos realizados	1,2,3,4,5	40%
3. Desarrollar un programa informático que sea correcto desde el punto de vista de la metodología de la Programación	1,2,3,4,5	35%
4. Participar activamente en las tutorías ECTS y en los seminarios.	1,2,3,4,5	5% (NR)
5. Asistir a seminarios	1,2,3,4,5	5% (NR)

<i>Actividades e instrumentos de evaluación (1er cuatrimestre)</i>		
Seminarios y Tutorías ECTS	• Prueba objetiva de las prácticas. Examen tipo test	15%
	• Elaboración de una práctica final	35%
	• Asistencia a seminarios	5%
	• Participación activa en las tutorías ECTS	5%
Grupo Grande	• Examen final	40%

6. Bibliografía

<i>Bibliografía de apoyo seleccionada</i>	
1.1	Programación orientada a objetos. Roberto Rodríguez, Encarna Sosa y Alvaro Prieto.
1.2	Programación en C++. Luis Joyanes.
1.3	Como programar en C++ . H. M. Deitel.
1.4	Resolución de problemas con C++. Savitch.
1.5	El lenguaje de programación C++. Stroustrup, B.
1.6	Aprendiendo C++ para linux en 21 días. Jesse Liberty y David B. Horvath.
1.7	Thinking in C++. Bruce Eckel.
1.8	Estructura de Datos, Algoritmos y Programación Orientada a Objetos. Gregory L. Heileman. Ed. McGraw-Hill, 1998 Fundamentals of Data Structures in C++. E. Horowitz, S. Sahni, D. Mehta. Computer Science Press, 1997
<i>Bibliografía o documentación de lectura obligatoria*</i>	
1.9	Programación orientada a objetos. Roberto Rodríguez, Encarna Sosa y Alvaro Prieto.
1.10	Aprendiendo C++ para linux en 21 días. Jesse Liberty y David B. Horvath.
<i>Bibliografía o documentación de ampliación, sitios web...*</i>	
1.11	Thinking in C++. Bruce Eckel. http://www.mindview.net/Books/TICPP/
1.12	Zator: http://www.zator.com/Cpp/
1.13	Aprenda C++ como si estuviera en primero. Manual en PDF.

i

Códigos.-

□ *CET: Competencias Específicas del Título* (véase el apartado de Contextualización curricular)

ii

Tipos de actividades: GG (Grupo Grande); S (Seminario o Laboratorio); Tut (Tutoría ECTS); No presenciales (NP); C-E, I (Coordinación o evaluación); T, II (Teórica de carácter expositivo o de aprendizaje a partir de documentos); T, III (Teórica de discusión); P, IV (Prácticas basadas en la solución de problemas); P, V (Prácticas basadas en la observación, experimentación, aplicación de destrezas, estudio de casos...); P, VI (Prácticas con proyectos o trabajos dirigidos); T-P, VII (Otras teórico-prácticas).

iii

D: Duración en sesiones de 1 hora de trabajo presencial o no presencial (considerando en cada hora 50-55 minutos de trabajo neto y 5-10 de descanso).