



DATOS IDENTIFICATIVOS

Programación I

Asignatura	Programación I			
Código	V05G300V01205			
Titulación	Grado en Ingeniería de Tecnologías de Telecomunicación			
Descriptores	Creditos ECTS	Selección	Curso	Cuatrimestre
	6	OB	1	1c
Lengua	Castellano			
Impartición	Gallego			
Departamento	Ingeniería telemática			
Coordinador/a	Rodríguez Hernández, Pedro Salvador			
Profesorado	Arriba Pérez, Francisco de García Palomares, Ubaldo Manuel Gil Solla, Alberto López Bravo, Cristina Pazos Arias, José Juan Rodríguez Hernández, Pedro Salvador Sousa Vieira, Estrella			
Correo-e	pedro.rodriguez@uvigo.es			
Web	http://fatic.uvigo.es			
Descripción general	El objetivo principal de esta asignatura es el desarrollo de capacidades de programación en un lenguaje de alto nivel. El paradigma de programación que se sigue es el de programación estructurada.			

Competencias

Código	
B4	CG4 Capacidad para resolver problemas con iniciativa, para la toma de decisiones, la creatividad, y para comunicar y transmitir conocimientos, habilidades y destrezas, comprendiendo la responsabilidad ética y profesional de la actividad del Ingeniero Técnico de Telecomunicación.
B9	CG9 Capacidad para trabajar en un grupo multidisciplinar y en un entorno multilingüe y de comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.
C6	CE6/T1 Capacidad para aprender de manera autónoma nuevos conocimientos y técnicas adecuados para la concepción, el desarrollo o la explotación de sistemas y servicios de telecomunicación.
C12	CE12/T7 Conocimiento y utilización de los fundamentos de la programación en redes, sistemas y servicios de telecomunicación.
D2	CT2 Concebir la Ingeniería en un marco de desarrollo sostenible.
D4	CT4 Favorecer el trabajo cooperativo, las capacidades de comunicación, organización, planificación y aceptación de responsabilidades en un ambiente de trabajo multilingüe y multidisciplinar, que favorezca la educación para la igualdad, para la paz y para el respeto de los derechos fundamentales.

Resultados de aprendizaje

Resultados previstos en la materia	Resultados de Formación y Aprendizaje
Expresar la solución de un problema sencillo mediante algoritmos, utilizando diseño descendente.	C12
Identificar los datos necesarios para resolver un problema y asociarlos a los tipos correspondientes en función de sus características (tamaño, rango, operadores que actúan sobre ellos).	C12
Codificar algoritmos sencillos a partir de los tres tipos básicos de sentencias: asignación, selección e iteración.	C12
Declarar y definir funciones haciendo un uso adecuado del paso de parámetros.	C12
Manejar las operaciones de entrada/salida y operar con ficheros.	C12
Definir y utilizar tipos de datos estructurados.	C12
Definir y gestionar estructuras de datos dinámicas (listas, pilas, colas y árboles).	C12

Crear y utilizar módulos y funciones de biblioteca dentro de un programa.	C6		
	C12		
Predecir el resultado de una secuencia de sentencias básicas, conocidos los datos de entrada.	C12		
Manejar herramientas básicas para el desarrollo de programas: editor de textos, compilador, enlazador, depurador y herramientas para la documentación.	C6		
Desarrollar proyectos software de pequeña envergadura siguiendo todas las fases: análisis de requisitos, diseño, construcción, evaluación y documentación.	B4 B9	C6 C12	D2 D4

Contenidos

Tema	
Tema 1: El algoritmo y los lenguajes de programación.	1. El algoritmo y sus distintas representaciones: diagrama de flujo, pseudocódigo, lenguaje natural. 2. Implementación del algoritmo mediante un lenguaje de programación. Paradigmas de programación: La programación modular y la programación estructurada. 3. El lenguaje C y la función main(). Código fuente y código objeto. El compilador y el intérprete. 4. Ejercicios de entrada/salida: interfaz persona-ordenador. Los ficheros de entrada/salida estándar: stdin, stdout. La directiva #include. Funciones de biblioteca.
Tema 2: La gramática y elementos básicos del lenguaje C.	1. El alfabeto. Derivaciones recursivas de secuencias sintácticamente válidas. Identificadores, números. Constantes simbólicas: La directiva #define y macros. Uso del calificador const. 2. Variables y sus atributos: nombre, valor, dirección, tipos. La variable puntero. Declaración de variables simples y punteros: los operadores de dirección & y de referencia * 3. El operador sizeof. Operadores aritméticos. El operador de asignación. Conversión automática de tipos y mediante el operador cast. 4. Notación sintáctica para expresiones e instrucciones. Instrucción simple e instrucción compuesta.
Tema 3: Instrucciones secuenciales, iterativas y de selección	1. Evaluación de expresiones con operadores relacionales y operadores booleanos. 2. Instrucciones de selección: switch, if, if anidado. El operador ternario (? :). 3. Las instrucciones iterativas y su importancia en la programación modular: while, do while y for. Instrucciones break y continue.
Tema 4: Variables indexadas	1. Declaración de tipo de variables indexadas (Arrays). Asignación de memoria para arrays multidimensionales. 2. Arrays unidimensionales y punteros: aritmética de punteros. Arrays de caracteres: El carácter de fin de cadena.. 3. Los arrays de longitud variable en el estándar C99. 4. Asignación dinámica de memoria a arrays de una y 2 dimensiones: las funciones malloc(), calloc(), realloc().
Tema 5: Funciones	1. Declaración y definición de funciones. Variables locales, estáticas y globales. Valor de retorno de una función. 2. Parámetros actuales y parámetros formales. Paso de parámetros por valor y por referencia: uso de punteros. Paso de parámetros por línea de comando a la función main(). 3. Elaboración y uso de bibliotecas de funciones. Funciones de biblioteca que gestionan arrays de caracteres. 4. Compilación modular. Las directivas condicionales en un fichero de cabecera. 5. Funciones recursivas: ventajas y desventajas.
Tema 6: Variables tipo struct	1. Variables tipo struct: declaración global. Campos de un struct. Punteros a struct. Los operadores . (punto) y -> (flecha). 2. El struct y un puntero a struct como parámetro y valor de retorno de una función. 3. El typedef con declaraciones no triviales. 4. Estructuras más complejas: struct anidados, array de struct. 5. Gestión dinámica en creación de listas lineales, listas circulares, árboles. 6. Inserción y remoción de variables en una lista.

1. Ficheros de texto: funciones `fopen()`, `fclose()`.
2. Diferentes funciones de entrada/salida para ficheros: `fprintf()`, `fscanf()`, `fgets()`, `feof()`.
3. Las funciones con acceso directo al fichero.
4. Manejo de la información entre ficheros y listas.
5. Estructura de los nodos en listas enlazadas de forma simple.
6. Paso de ficheros a lista y viceversa.

Planificación

	Horas en clase	Horas fuera de clase	Horas totales
Actividades introductorias	2	0	2
Lección magistral	22	27	49
Prácticas de laboratorio	12	12	24
Aprendizaje basado en proyectos	10	28	38
Práctica de laboratorio	5	15	20
Otras	5	10	15
Informe de prácticas	0	2	2

*Los datos que aparecen en la tabla de planificación son de carácter orientativo, considerando la heterogeneidad de alumnado

Metodologías

	Descripción
Actividades introductorias	Introducción a la materia tanto en su componente teórica como práctica.
Lección magistral	Presentación por parte del profesorado del temario de la materia. Estas sesiones incluirán la realización de trabajos y la realización de programas por parte del alumnado. Con esta metodología se trabajan las competencias CE12 y CT2.
Prácticas de laboratorio	A lo largo de la primera parte de cuatrimestre, los/as estudiantes codificarán, desarrollarán y documentarán sencillos programas, guiados por el profesorado. En algunas de ellas se pedirá la entrega de informes para su evaluación. Con esta metodología se trabajan las competencias CG4, CE12 y CT2.
Aprendizaje basado en proyectos	En la segunda parte del laboratorio se propone al alumnado la realización de un pequeño proyecto. Este proyecto se realiza en las últimas sesiones prácticas del cuatrimestre e incluirá actividades individuales y en grupo. Con esta metodología se trabajan las competencias CG4, CG9, CE6, CE12, CT2 y CT4.

Atención personalizada

Metodologías	Descripción
Lección magistral	El profesorado de la asignatura proporcionará atención individual y personalizada al alumnado durante el curso, solucionando sus dudas y preguntas. Las dudas se atenderán de forma presencial (durante la propia sesión magistral, o durante el horario establecido para las tutorías). El horario de tutorías se establecerá al principio del curso y se publicará en la página web de la asignatura.
Prácticas de laboratorio	El profesorado de la materia proporcionará atención individual y personalizada al alumnado durante el curso, solucionando sus dudas y preguntas. Así mismo, el profesorado orientará y guiará al alumnado durante la realización de las tareas que tienen asignadas en las prácticas de laboratorio. Las dudas se atenderán de forma presencial (durante las prácticas, o durante el horario establecido para tutorías). El horario de tutorías se establecerá al principio del curso y se publicará en la página web de la asignatura.
Aprendizaje basado en proyectos	El profesorado de la materia proporcionará atención individual y personalizada al alumnado durante el curso, solucionando sus dudas y preguntas. Asimismo, el profesorado orientará y guiará a los alumnos durante la realización de las tareas que tienen asignadas para la realización del proyecto. Las dudas se atenderán de forma presencial (durante las propias sesiones de seguimiento, o durante el horario establecido para las tutorías). El horario de tutorías se establecerá al principio del curso y se publicará en la página web de la asignatura.

Evaluación

	Descripción	Calificación	Resultados de Formación y Aprendizaje
Aprendizaje basado en proyectos	Se desarrollará un proyecto en las últimas semanas del curso, y se entregará el código C que lo implementa. La evaluación del proyecto es individual, y se realizará mediante la prueba práctica final.	25	B4 C6 D4 B9 C12

Práctica de laboratorio	Cada 4 semanas, se realizará una prueba en el laboratorio de forma individual que consistirá en la realización de un programa en el ordenador. Se hará una prueba práctica final que evaluará la realización de las prácticas de laboratorio y del proyecto.	20	B4	C12	
Otras	Cada 4 semanas, se hará un examen de teoría que puede contener: - cuestiones de respuesta corta - cuestiones tipo test - resolución de problemas y/o ejercicios Este examen evaluará, de forma individual, el conocimiento de los conceptos introducidos en las sesiones magistrales. Se hará una prueba teórica final sobre todos los contenidos de la materia.	50	B4	C12	
Informe de prácticas	Tras la segunda semana de desarrollo del proyecto, se entregará el pseudocódigo o diagrama de flujo que describa su diseño. Al final, se entregará una memoria con la documentación del proyecto, que se evaluará de forma individual.	5	B4	C12	D4

Otros comentarios sobre la Evaluación

A continuación se muestra la **planificación de la asignatura por temas**, indicando el momento estimado de los **hitos de evaluación más importantes** (las fechas de las pruebas parciales teóricas y prácticas son tentativas: la planificación de las diferentes pruebas de evaluación intermedia se aprobará en una Comisión Académica de Grado (CAG) y estará disponible al principio del cuatrimestre).

- Semana 1: Introducción de teoría + Temas 1 y 2
- Semana 2: Tema 3 | Introducción de práctica + Práctica 1
- Semana 3: Temas 3 y 4 | Práctica 2
- Semana 4: Tema 4 + **Prueba Teórica 1** (PT1) | **Prueba Práctica 1** (PP1)
- Semana 5: Tema 4 | Práctica 3
- Semana 6: Tema 5 | Práctica 4
- Semana 7: Tema 5 | Práctica 5
- Semana 8: Tema 5 + **Prueba Teórica 2** (PT2) | **Prueba Práctica 2** (PP2)
- Semana 9: Temas 5 y 6 | Práctica 6
- Semana 10: Tema 6 | Finalización de Prácticas + Proyecto (1h)
- Semana 11: Tema 6 | Proyecto (2h) + Entrega del diseño del proyecto (seudocódigo o diagrama de flujo)
- Semana 12: Tema 7 + **Prueba Teórica 3** (PT3) | Proyecto (1h) + **Prueba Práctica 3** (PP3)
- Semana 13: Tema 7 | Proyecto (2h)
- Semana 14: Proyecto (2h)
- Previo al período de exámenes, entrega del proyecto: codificación y memoria de documentación
- Período de exámenes: **Prueba Teórica Final** (PTF) | **Prueba Práctica Final** (PPF)

Siguiendo las directrices propias de la titulación, se ofrece al alumnado que curse esta materia dos sistemas de evaluación: **evaluación continua y evaluación única**.

La decisión de optar por la evaluación única deberá ser tomada como muy tarde la semana antes de aquélla en la que se realiza la Prueba Práctica 2 (PP2).

Para aprobar la asignatura siguiendo el sistema de **evaluación continua**, es necesario obtener una nota final (NFC) igual o superior a 5.

La nota final por evaluación continua (que se calculará como la media geométrica ponderada de la Nota Teórica, la Nota Práctica y la Nota del Proyecto) vendrá dada por la siguiente expresión:

$$NFC = NTC^{0.5} \cdot NPC^{0.2} \cdot NPR^{0.3}$$

Donde:

- Nota Teórica por Evaluación Continua: $NTC = 0.1 \cdot PT1 + 0.1 \cdot PT2 + 0.2 \cdot PT3 + 0.6 \cdot PTF$
- Nota Práctica por Evaluación Continua: $NPC = 0.25 \cdot PP1 + 0.25 \cdot PP2 + 0.5 \cdot PP3$
- Nota del Proyecto: $NPR = 0.9 \cdot PPF + 0.1 \cdot PDD$

La Prueba Teórica Final (PTF) es un examen que puede contener cuestiones de respuesta corta y/o tipo test y/o de resolución de problemas y/o ejercicios. Evalúa el conocimiento de los contenidos introducidos en las sesiones magistrales. La Prueba Práctica Final (PPF) evalúa el proyecto entregado. Si bien el proyecto se desarrolla en grupo, se evalúa de forma

individual. De manera indirecta, la PPF también evalúa el conocimiento de los contenidos introducidos en las sesiones magistrales y en las prácticas de laboratorio.

La **Prueba de Diseño y Documentación (PDD)** evalúa la calidad del pseudocódigo o diagrama de flujo que describe el proyecto (entregado en la semana 11), y la memoria de documentación del proyecto entregada antes del periodo de exámenes.

Nótese que la aplicación de la media geométrica implica que no es posible aprobar la asignatura si alguna de las notas (NPC, NTC o NPR) es cero.

La evaluación continua consta de las tareas que se detallan en esta guía y no son recuperables, es decir, si un/a alumno/a no puede cumplirlas en el plazo estipulado, el profesorado no tiene la obligación de repetírselas.

Antes de la realización o entrega de cada tarea se indicará la fecha y procedimiento de revisión de las calificaciones obtenidas. El alumnado tendrá opción a conocer la calificación de cada tarea y revisar la corrección en un plazo aproximado de 2 semanas.

Para aprobar la asignatura en el sistema de **evaluación única**, será necesario obtener una nota final (NFU) igual o superior a 5.

Esta modalidad consistirá en las mismas pruebas que la de evaluación continua (aunque con distinto peso en la nota final), es decir una prueba con cuestiones de respuesta corta y/o tipo test y/o de resolución de problemas y/o ejercicios (Prueba Teórica Final, PTF) y una prueba en el laboratorio que evaluará el proyecto (Prueba Práctica Final, PPF). La nota final por evaluación única (que se calculará como la media geométrica ponderada entre la nota del proyecto y la nota de teoría) vendrá dada por la siguiente expresión:

$$NFU = PTF^{0.5} * NPR^{0.5}$$

A todos el alumnado que concurra al examen final de la asignatura, se le calcularán ambas notas: la **nota final por evaluación continua (NFC)** y la **nota final por evaluación única (NFU)**. La nota final que se le otorgará será la mayor de ambas.

La calificación será de "No Presentado" en los siguientes casos:

- En caso de optar por la evaluación continua, sólo si no realiza ninguna prueba después de la Prueba Práctica 1 (PP1).
- En caso de optar por la evaluación única, sólo si no realiza ninguna de las pruebas finales (PTF y PPF).

En la evaluación en segunda oportunidad, para aprobar la asignatura, será necesario obtener una nota final (NFS) igual o superior a 5.

En esta segunda oportunidad, se realizará una prueba con cuestiones de respuesta corta y/o tipo test y/o de resolución de problemas y/o ejercicios (Prueba Teórica en Segunda Oportunidad, PTS) y una prueba en el laboratorio que evaluará el proyecto (Prueba Práctica en Segunda Oportunidad, PPS). La nota final por evaluación en segunda oportunidad (que se calculará como la media geométrica ponderada entre la nota del proyecto y la nota de teoría) vendrá dada por la siguiente expresión:

$$NFS = NTS^{0.5} * NPS^{0.5}$$

Donde:

- Nota Teórica por Evaluación en segunda oportunidad (NTS): si el/la alumno/a se presenta a la Prueba Teórica en Segunda Oportunidad, NTS será la nota obtenida en dicha prueba:

$$NTS = PTS$$

Si no, NTS será la nota de teoría obtenida en la evaluación en primera oportunidad.

- Nota Práctica por Evaluación en segunda oportunidad (NPS): si el/la alumno/a se presenta a la Prueba Práctica en Segunda Oportunidad, NPS será la suma ponderada de la nota obtenida en dicha prueba y la nota obtenida en la prueba de diseño y documentación

$$NPS = 0.9*PPS + 0.1*PDD$$

Si no, NPS será la nota de prácticas obtenida en la evaluación en primera oportunidad.

En la convocatoria extraordinaria de fin de carrera, para aprobar la asignatura, será necesario obtener una nota final (NFG) igual o superior a 5.

En esta convocatoria extraordinaria, se realizará una prueba con cuestiones de respuesta corta y/o tipo test y/o de resolución de problemas y/o ejercicios (Prueba Teórica de Fin de Carrera, PTG) y una prueba en el laboratorio que evaluará el proyecto (Prueba Práctica de Fin de Carrera, PPG). La nota final por evaluación en la convocatoria extraordinaria de fin de

carrera (que se calculará como la media geométrica ponderada entre la nota del proyecto y la nota de teoría) vendrá dada por la siguiente expresión:

$$NFG = PTG^{0.5} * PPG^{0.5}$$

La calificación obtenida en cualquiera de las tareas evaluables será válida tan solo para el curso académico en el que se realicen, es decir, no se guarda ninguna nota de un curso para el siguiente.

En caso de detección de plagio en alguno de los trabajos/pruebas realizadas la calificación será de suspenso (0) y el profesorado comunicará a la dirección de la escuela el asunto para que tome las medidas que considere oportunas.

Fuentes de información

Bibliografía Básica

Brian W. Kernighan & Dennis M. Ritchie, **The C Programming Language**, 1995, Prentice Hall, 1983

Brian W. Kernighan & Dennis M. Ritchie, **El Lenguaje de Programación C**, 1995, Prentice Hall, 1983

Manuel Caeiro Rodríguez, Enrique Costa Montenegro, Ubaldo García Palomares, Cristina López Bravo, J, **Practicar Programación en C**, 2014,

Bibliografía Complementaria

Ignacio Alvarado Aldea, Jose María Maestre Torreblanca, Carlos Vivas Venegas, Ascensión Zafra Cabeza, **100 Problemas Resueltos de Programación en Lenguaje C para Ingeniería**, 2017, Paraninfo, 2017

Stephen G. Kochan, **Programming in C**, 2014, 2005

Osvaldo Cairo Battistuti, **Fundamentos de Programación**, 2006,

José Rafael García-Bermejo Giner, **Programación Estructurada en C**, 2008,

James L. Antonakos & Kenneth C. Mansfield Jr., **Programación Estructurada en C**, 2004, 1997

Jorge A. Villalobos S. & Rubby Casallas G., **Fundamentos de Programación: Aprendizaje Activo Basado en Casos**, 2006,

Recomendaciones

Asignaturas que continúan el temario

Programación II/V05G300V01302

Asignaturas que se recomienda haber cursado previamente

Informática: Arquitectura de ordenadores/V05G300V01103

Otros comentarios

La asignatura Programación II es una continuación de esta asignatura en el segundo curso.