



DATOS IDENTIFICATIVOS

Programación I

Asignatura	Programación I			
Código	V05G300V01205			
Titulación	Grado en Ingeniería de Tecnologías de Telecomunicación			
Descriptores	Creditos ECTS	Seleccione	Curso	Cuatrimestre
	6	OB	1	2c
Lengua Impartición	Castellano			
Departamento	Ingeniería telemática			
Coordinador/a	Rodríguez Hernández, Pedro Salvador			
Profesorado	García Palomares, Ubaldo Manuel Pazos Arias, José Juan Ramos Cabrer, Manuel Rodríguez Hernández, Pedro Salvador			
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.
Tema 7: Ficheros	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
Sesión magistral	24	24	48
Prácticas de laboratorio	12	14	26
Proyectos	8	24	32
Pruebas prácticas, de ejecución de tareas reales y/o simuladas.	5	15	20
Otras	5	15	20
Informes/memorias 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.
Sesión 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	En el laboratorio se expondrá el desarrollo de prácticas guiadas y la realización de programas. Estas prácticas tienen lugar a lo largo de la primera parte del cuatrimestre. En algunas de ellas se pedirá la entrega de informes para su evaluación. Con esta metodología se trabajan las competencias CG4, CE12 y CT2.
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
Sesión 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.
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
Proyectos	Se desarrollará un proyecto en las últimas semanas del curso, y se entregará el código C que lo implementa. La evaluación del proyecto se realizará mediante la prueba práctica final.	25	B4 C6 D4 B9 C12
Pruebas prácticas, de ejecución de tareas reales y/o simuladas.	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 que puede contener: - cuestiones de respuesta corta - cuestiones tipo test - resolución de problemas y/o ejercicios Este examen evaluará 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	
Informes/memorias 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.	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**:

- 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)**

Seguendo las directrices propias de la titulación, se ofrece al alumnado que curse esta materia dos sistemas de evaluación: **evaluación continua** y **evaluación al final del cuatrimestre**.

La decisión de optar por la evaluación al final del cuatrimestre deberá ser tomada como muy tarde la semana antes de aquella en la que se realiza la Prueba Práctica 2 (PP2).

Para aprobar la asignatura siguiendo el sistema de **evaluación continua**, es necesario cumplir dos condiciones:

1. haber entregado el proyecto (diseño, codificación y documentación) realizado en las semanas 10 a 14 del curso, y
2. obtener una nota final (NFC) igual o superior a 5.

La nota final por evaluación continua (que se calculará como la media armónica entre la suma de las pruebas prácticas y la suma de las pruebas teóricas) vendrá dada por la siguiente expresión:

$$NFC = (2 \cdot NTC \cdot NPC) / (NTC + NPC)$$

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: $NPC = 0.1 \cdot PP1 + 0.1 \cdot PP2 + 0.2 \cdot PP3 + 0.5 \cdot PPF + 0.1 \cdot PDD$

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 armónica implica que no es posible aprobar la asignatura si alguna de las notas (NPC o NTC) es inferior a 3.3.

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 al final del cuatrimestre**, será necesario:

1. haber entregado un proyecto (diseño, codificación y documentación) como el realizado por el alumnado de evaluación continua en las semanas 10 a 14 del curso, y
2. obtener una nota final (NFF) 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 final (que se calculará como la media armónica entre la nota de práctica y la nota de teoría) vendrá dada por la siguiente expresión:

$$NFF = (2 \cdot NTF \cdot NPF) / (NTF + NPF)$$

Donde:

- Nota Teórica por Evaluación Final: $NTF = PTF$
- Nota Práctica por Evaluación Final: $NPF = 0.9 \cdot PPF + 0.1 \cdot PDD$

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 final (NFF)**. 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 al final del cuatrimestre, sólo si no realiza ninguna de las pruebas finales (PTF y PPF).

En las convocatorias extraordinarias, para aprobar la asignatura, será necesario:

1. haber entregado un proyecto (diseño, codificación y documentación) como el realizado por el alumnado de evaluación continua en las semanas 10 a 14 del curso, y
2. obtener una nota final igual o superior a 5.

En estas convocatorias, se realizará una prueba con cuestiones de respuesta corta y/o tipo test y/o de resolución de problemas y/o ejercicios (Prueba Teórica Extraordinaria, PTE) y una prueba en el laboratorio que evaluará el proyecto (Prueba Práctica Extraordinaria, PPE). La nota final por evaluación extraordinaria (que se calculará como la media armónica entre la nota de práctica y la nota de teoría) vendrá dada por la siguiente expresión:

$$NFE = (2 \cdot NTE \cdot NPE) / (NTE + NPE)$$

Donde:

- Nota Teórica por Evaluación Extraordinaria (NTE): si el/la alumno/a se presenta a la Prueba Teórica Extraordinaria, NTE será la nota obtenida en dicha prueba:

$$NTE = PTE$$

Si no, NTE será la nota de teoría obtenida en la evaluación ordinaria.

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

$$NPE = 0.9 \cdot PPE + 0.1 \cdot PDD$$

Si no, NPE será la nota de prácticas obtenida en la evaluación ordinaria.

En todas las convocatorias, en cualquiera de las modalidades de cálculo de la nota, en el caso de incumplir la condición necesaria de presentar el proyecto, la nota final (calculada según la correspondiente fórmula) estará acotada superiormente a un máximo de 4.5 puntos.

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

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

Brian W. Kernighan & Dennis M. Ritchie, **El Lenguaje de Programación C**, 1986 (reimpreso en 1995),

Bibliografía Complementaria

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**, 1997 (reimpreso en 2004),

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.
