



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	Castellano			
Impartición				
Departamento	Ingeniería telemática			
Coordinador/a	Rodríguez Hernández, Pedro Salvador			
Profesorado	García Palomares, Ubaldo Manuel Rodríguez Hernández, Pedro Salvador Santos Suárez, José Manuel			
Correo-e	pedro.rodriguez@uvigo.es			
Web	http://faitic.uvigo.es			
Descripción	El objetivo principal de esta asignatura es el desarrollo de capacidades de programación en un lenguaje de general 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.	<ol style="list-style-type: none"> 1. El algoritmo y sus distintas representaciones: diagrama de flujo, pseudo código, 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.	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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: Funciones: Introducción	<ol style="list-style-type: none"> 1. Aritmética de punteros. Arrays y punteros. Asignación dinámica de memoria a arrays de 1 y 2 dimensiones: las funciones malloc(), calloc(), realloc(). 2. Arrays de caracteres: El carácter fin de cadena. Funciones de biblioteca que gestionan arrays de caracteres. 3. Declaración y definición de funciones. Variables locales en una instrucción compuesta. Paso de parámetros por valor y por referencia: uso de punteros. El retorno de una función. 4. Variables estáticas y variables globales.
Tema 5: Funciones: Casos especiales	<ol style="list-style-type: none"> 1. Paso de argumentos en línea de comando: argc y argv. 2. Funciones recursivas: ventajas y desventajas. 3. Elaboración y uso de bibliotecas de funciones. Las directivas condicionales en un fichero de cabecera. 4. Funciones que retornan direcciones.
Tema 6: Variables tipo struct	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 1. Ficheros de texto: funciones fopen(), fclose(). 2. Diferentes funciones de entrada/salida para ficheros: fprintf(), fscanff(), 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	16	28
Proyectos	8	24	32
Pruebas prácticas, de ejecución de tareas reales y/o simuladas.	5	15	20
Resolución de problemas y/o ejercicios	5	15	20

*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 de los alumnos. 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	Los estudiantes tendrán ocasión de acudir a tutorías personalizadas en el despacho del profesor en el horario que los profesores establecerán a tal efecto a principio de curso y que se publicará en la página de la asignatura. Se prevén tutorías personalizadas para la resolución de dudas sobre los conceptos introducidos en la sesión magistral y las actividades realizadas en las prácticas de laboratorio y en el proyecto.
Prácticas de laboratorio	Los estudiantes tendrán ocasión de acudir a tutorías personalizadas en el despacho del profesor en el horario que los profesores establecerán a tal efecto a principio de curso y que se publicará en la página de la asignatura. Se prevén tutorías personalizadas para la resolución de dudas sobre los conceptos introducidos en la sesión magistral y las actividades realizadas en las prácticas de laboratorio y en el proyecto.
Proyectos	Los estudiantes tendrán ocasión de acudir a tutorías personalizadas en el despacho del profesor en el horario que los profesores establecerán a tal efecto a principio de curso y que se publicará en la página de la asignatura. Se prevén tutorías personalizadas para la resolución de dudas sobre los conceptos introducidos en la sesión magistral y las actividades realizadas en las prácticas de laboratorio y en el proyecto.

Evaluación

	Descripción	Calificación	Resultados de Formación y Aprendizaje
Proyectos	Se desarrollará un proyecto en las últimas semanas del curso, y se entregará un informe. La evaluación del proyecto se realizará mediante la prueba final práctica.	30	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 final que evaluará la realización de las prácticas de laboratorio y del proyecto.	20	C6 C12
Resolución de problemas y/o ejercicios	Cada 4 semanas, se realizará un examen de resolución de ejercicios, que evaluará el conocimiento de los conceptos introducidos en las sesiones magistrales. Se hará una prueba final sobre todos los contenidos de la materia.	50	B4 C12

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: Tema 1/2

- Semana 2: Tema 3 - Práctica 1
- Semana 3: Tema 3 - Práctica 1/2
- Semana 4: Tema 4 - **Prueba Teórica 1** (PT1) - **Prueba Práctica 1** (PP1)
- Semana 5: Tema 4 - Práctica 2/3
- Semana 6: Tema 4 - Práctica 3/4
- Semana 7: Tema 5 - Práctica 4/5
- Semana 8: Tema 5 - **Prueba Teórica 2** (PT2) - **Prueba Práctica 2** (PP2)
- Semana 9: Tema 5/6 - Práctica 5/6
- Semana 10: Tema 6 - Práctica 6 - Proyecto (1h)
- Semana 11: Tema 6 - Proyecto (2h)
- Semana 12: Tema 7 - Proyecto (1h) - **Prueba Teórica 3** (PT3) - **Prueba Práctica 3** (PP3)
- Semana 13: Tema 7 - Proyecto (2h)
- Semana 14: Proyecto (2h)
- Período de exámenes: **Prueba Final teórica** (PFT) - **Prueba Final práctica** (PFP)

Seguindo las directrices propias de la titulación, se ofrece a los alumnos que cursen 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 aquélla en la que se realiza en el laboratorio la Prueba Teórica 2 (PT2).

Para aprobar la asignatura siguiendo el sistema de **evaluación continua**, es necesario cumplir dos condiciones: haber entregado el proyecto realizado en las semanas 10 a 14 del curso, y obtener una nota final (NF) igual o superior a 5. La nota final (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:

$$NF = (2*NT*NP)/(NT+NP)$$

Donde:

- $NP = 0.1*PP1+0.1*PP2+0.2*PP3+0.6*PFP$
- $NT = 0.1*PT1+0.1*PT2+0.2*PT3+0.6*PFT$

La Prueba Final Teórica (PFT) evalúa el conocimiento de los contenidos introducidos en las sesiones magistrales.

La Prueba Final Práctica (PFP) evalúa el proyecto entregado. De manera indirecta, también evalúa el conocimiento de los contenidos introducidos en las sesiones magistrales y en las prácticas de laboratorio.

Nótese que la aplicación de la media armónica implica que no es posible aprobar la asignatura si alguna de las notas (NP o NT) 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 alumno no puede cumplirlas en el plazo estipulado, el profesor 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 haber entregado un proyecto como el realizado por los alumnos de evaluación continua en las semanas 10 a 14 del curso, y habrá que obtener por lo menos una nota final igual o superior a 5. Este sistema consistirá en una prueba de resolución de ejercicios (parte teórica) y una prueba en el laboratorio que evaluará el proyecto (parte práctica). La nota final (que se calculará como la media armónica entre la nota de práctica y la nota de teoría) vendrá dada por la siguiente expresión:

$$NF = (2*NT*NP)/(NT+NP)$$

Donde:

- $NP = PFP$
- $NT = PFT$

A todos los alumnos que concurran al examen final de la asignatura, se les calcularán ambas notas: la **nota de evaluación continua** y la **nota de evaluación al final de cuatrimestre**. 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 (PFT y PFP).

En las convocatorias extraordinarias, para aprobar la asignatura, será necesario haber entregado un proyecto como el realizado por los alumnos de evaluación continua en las semanas 10 a 14 del curso, y habrá que obtener por lo menos una nota final igual o superior a 5. En estas convocatorias, se realizará una prueba de resolución de ejercicios (parte teórica) y una prueba en el laboratorio que evaluará el proyecto (parte práctica). La nota final (que se calculará como la media armónica entre la nota de práctica y la nota de teoría) vendrá dada por la siguiente expresión:

$$NF = (2*NT*NP)/(NT+NP)$$

Donde:

- Si el alumno se presenta a la prueba de resolución de ejercicios, NT será la nota obtenida en dicha prueba. Si no, NT será la nota de teoría obtenida en la evaluación ordinaria.
- Si el alumno se presenta a la prueba en el laboratorio, NP será la nota obtenida en dicha prueba. Si no, NP será la nota de prácticas obtenida en la evaluación ordinaria.

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 los profesores comunicarán a la dirección de la escuela el asunto para que tome las medidas que considere oportunas.

Fuentes de información

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

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

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

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

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,

Recursos Web

- <http://www.Cprogramming.com>
- José R. García-Bermejo Giner: http://maxus.fis.usal.es/FICHAS_C.WEB/11xx_PAGS/11xx.html

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.