



DATOS IDENTIFICATIVOS

Programación I

Materia	Programación I			
Código	V05G300V01205			
Titulación	Grao en Enxeñaría de Tecnoloxías de Telecomunicación			
Descritores	Creditos ECTS	Sinale	Curso	Cuadrimestre
	6	OB	1	2c
Lingua de impartición	Castelán			
Departamento	Enxeñaría 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			
Descrición xeral	O obxectivo principal desta materia é o desenvolvemento de capacidades de programación nunha linguaxe de alto nivel. O paradigma de programación que se segue é o de programación estruturada.			

Competencias

Código	
B4	CG4 Capacidade para resolver problemas con iniciativa, para a toma de decisións, a creatividade, e para comunicar e transmitir coñecementos, habilidades e destrezas, comprendendo a responsabilidade ética e profesional da actividade do Enxeñeiro Técnico de Telecomunicación.
B9	CG9 Capacidade para traballar nun grupo multidisciplinar e nunha contorna multilingüe e de comunicar, tanto por escrito como de forma oral, coñecementos, procedementos, resultados e ideas relacionadas coas telecomunicacións e a electrónica.
C6	CE6/T1 Capacidade para aprender de xeito autónomo novos coñecementos e técnicas axeitados para a concepción, o desenvolvemento ou a explotación de sistemas e servizos de telecomunicación.
C12	CE12/T7 Coñecemento e utilización dos fundamentos da programación en redes, sistemas e servizos de telecomunicación.
D2	CT2 Concibir a Enxeñaría no marco do desenvolvemento sostible.
D4	CT4 Favorecer o traballo cooperativo, as capacidades de comunicación, organización, planificación e aceptación de responsabilidades nun ambiente de traballo multilingüe e multidisciplinar, que favoreza a educación para a igualdade, para a paz e para o respecto dos dereitos fundamentais.

Resultados de aprendizaxe

Resultados previstos na materia	Resultados de Formación e Aprendizaxe
Expresar a solución dun problema sinxelo mediante algoritmos, utilizando deseño descendente.	C12
Identificar os datos necesarios para resolver un problema e asocialos aos tipos correspondentes en función das súas características (tamaño, rango, operadores que actúan sobre eles).	C12
Codificar algoritmos sinxelos a partir do tres tipos básicos de sentenzas: asignación, selección e iteración.	C12
Declarar e definir funcións facendo un uso adecuado do paso de parámetros.	C12
Manexar as operacións de entrada/saída e operar con ficheiros.	C12
Definir e utilizar tipos de datos estruturados.	C12
Definir e xestionar estruturas de datos dinámicas (listas, pilas, colas e árbores).	C12
Crear e utilizar módulos e funcións de biblioteca dentro dun programa.	C6 C12
Predicir o resultado dunha secuencia de sentenzas básicas, coñecidos os datos de entrada.	C12
Manexar ferramentas básicas para o desenvolvemento de programas: editor de textos, compilador, enlazador, depurador e ferramentas para a documentación.	C6

Desenvolver proxectos software de pequena envergadura seguindo todas as fases: análise de requisitos, deseño, construción, avaliación e documentación.	B4 B9	C6 C12	D2 D4
--	----------	-----------	----------

Contidos

Tema	
Tema 1: O algoritmo e as linguaxes de programación.	<ol style="list-style-type: none"> O algoritmo e as súas distintas representacións: diagrama de fluxo, pseudo código, linguaxe natural. Implementación do algoritmo mediante unha linguaxe de programación. Paradigmas de programación: A programación modular e a programación estruturada. A linguaxe C e a función main(). Código fonte e código obxecto. O compilador e o intérprete. Exercicios de entrada/saída: interfaz persoa-ordenador. Os ficheiros de entrada/saída estándar: stdin, stdout. A directiva #include . Funcións de biblioteca.
Tema 2: A gramática e elementos básicos da linguaxe C.	<ol style="list-style-type: none"> O alfabeto. Derivacións recursivas de secuencias sintácticamente válidas. Identificadores, números. Constantes simbólicas: A directiva #define e macros. Uso do cualificador const. Variables e os seus atributos: nome, valor, dirección, tipos. A variable punteiro. Declaración de variables simples e punteiros: os operadores de dirección & e de referencia * O operador sizeof. Operadores aritméticos. O operador de asignación. Conversión automática de tipos e mediante o operador cast. Notación sintáctica para expresións e instrucións. Instrución simple e instrución composta.
Tema 3: Instrucións secuenciais, iterativas e de selección	<ol style="list-style-type: none"> Avaliación de expresións con operadores relacionais e operadores booleanos. Instrucións de selección: switch, if, if anidado. O operador ternario (?:) As instrucións iterativas e a súa importancia na programación modular: while, do while e for. Instrucións break e continue.
Tema 4: Funcións: Introducción	<ol style="list-style-type: none"> Aritmética de punteiros. Arrays e punteiros. Asignación dinámica de memoria a arrays de 1 e 2 dimensións: as funcións malloc(), calloc(), realloc(). Arrays de caracteres: O carácter fin de cadea. Funcións de biblioteca que xestionan arrays de caracteres. Declaración e definición de funcións. Variables locais nunha instrución composta. Paso de parámetros por valor e por referencia: uso de punteiros. O retorno dunha función. Variables estáticas e variables globais.
Tema 5: Funcións: Casos especiais	<ol style="list-style-type: none"> Paso de argumentos en liña de comando: argc e argv. Funcións recursivas: vantaxes e desvantaxes. Elaboración e uso de bibliotecas de funcións. As directivas condicionais nun ficheiro de cabeceira. Funcións que retornan direccións.
Tema 6: Variables tipo struct	<ol style="list-style-type: none"> Variables tipo struct: declaración global. Campos dun struct. Punteiros a struct. Os operadores . (punto) e -> (frecha). O struct e un punteiro a struct como parámetro e valor de retorno dunha función. O typedef con declaracións non triviais. Estruturas máis complexas: struct aniñados, array de struct. Xestión dinámica en creación de listas lineais, listas circulares, árbores. Inserción e remoción de variables nunha lista
Tema 7: Ficheiros	<ol style="list-style-type: none"> Ficheiros de texto: funcións fopen(), fclose(). Diferentes funcións de entrada/saída para ficheiros: fprintf (), fscanf(), fgets(), feof(). As funcións con acceso directo ao ficheiro. Manexo da información entre ficheiros e listas. Estrutura dos nodos en listas enlazadas de forma simple. Paso de ficheiros a lista e viceversa.

Planificación

	Horas na aula	Horas fóra da aula	Horas totais
Actividades introdutorias	2	0	2
Sesión maxistral	24	24	48
Prácticas de laboratorio	12	16	28
Proxectos	8	24	32

Probas prácticas, de execución de tarefas reais e/ou simuladas.	5	15	20
Resolución de problemas e/ou exercicios	5	15	20

*Os datos que aparecen na táboa de planificación son de carácter orientador, considerando a heteroxeneidade do alumnado.

Metodoloxía docente

	Descrición
Actividades introdutorias	Introdución á materia tanto na súa compoñente teórica como práctica.
Sesión maxistral	Presentación por parte do profesorado do temario da materia. Estas sesións incluírán a realización de traballos e a realización de programas por parte dos alumnos. Con esta metodoloxía trabállanse as competencias CE12 e CT2.
Prácticas de laboratorio	No laboratorio exporase o desenvolvemento de prácticas guiadas e a realización de programas. Estas prácticas teñen lugar ao longo da primeira parte do cuadrimestre. Nalgunhas delas pedirase a entrega de informes para a súa avaliación. Con esta metodoloxía trabállanse as competencias CG4, CE12 e CT2.
Proxectos	Na segunda parte do laboratorio propónse ao alumnado a realización dun pequeno proxecto. Este proxecto realízase nas últimas sesións prácticas do cuadrimestre e incluírá actividades individuais e en grupo. Con esta metodoloxía trabállanse as competencias CG4, CG9, CE6, CE12, CT2 e CT4.

Atención personalizada

Metodoloxías	Descrición
Sesión maxistral	Os estudantes terán ocasión de acudir a titorías personalizadas no despacho do profesor no horario que os profesores establecerán para ese efecto a principio de curso e que se publicará na páxina da materia. Dispóñense titorías personalizadas para a resolución de dúbidas sobre os conceptos introducidos na sesión maxistral e para as actividades realizadas nas prácticas de laboratorio e no proxecto.
Prácticas de laboratorio	Os estudantes terán ocasión de acudir a titorías personalizadas no despacho do profesor no horario que os profesores establecerán para ese efecto a principio de curso e que se publicará na páxina da materia. Dispóñense titorías personalizadas para a resolución de dúbidas sobre os conceptos introducidos na sesión maxistral e para as actividades realizadas nas prácticas de laboratorio e no proxecto.
Proxectos	Os estudantes terán ocasión de acudir a titorías personalizadas no despacho do profesor no horario que os profesores establecerán para ese efecto a principio de curso e que se publicará na páxina da materia. Dispóñense titorías personalizadas para a resolución de dúbidas sobre os conceptos introducidos na sesión maxistral e para as actividades realizadas nas prácticas de laboratorio e no proxecto.

Avaliación

	Descrición	Cualificación	Resultados de Formación e Aprendizaxe
Proxectos	Desenvolverase un proxecto nas últimas semanas do curso, e entregarase un informe. A avaliación do proxecto realízase mediante a proba final práctica.	30	B4 C6 D4 B9 C12
Probas prácticas, de execución de tarefas reais e/ou simuladas.	Cada 4 semanas, realízase unha proba no laboratorio de forma individual que consistirá na realización dun programa no computador. Farase unha proba final que avaliará a realización das prácticas de laboratorio e do proxecto.	20	C6 C12
Resolución de problemas e/ou exercicios	Cada 4 semanas, realízase un exame de resolución de exercicios, que avaliará o coñecemento dos conceptos introducidos nas sesións maxistrais. Farase unha proba final sobre todos os contidos da materia.	50	B4 C12

Outros comentarios sobre a Avaliación

A continuación móstrase a **planificación da materia por temas**, indicando o momento estimado dos **fitos de avaliación máis importantes**:

- Semana 1: Tema 1/2
- Semana 2: Tema 3 - Práctica 1

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

Seguindo as directrices propias da titulación, ofrécese aos alumnos que cursen esta materia dous sistemas de avaliación: **avaliación continua** e **avaliación ao final do cuadrimestre**.

A decisión de optar pola avaliación ao final do cuadrimestre deberá ser tomada como moi tarde a semana antes daquela na que se realiza no laboratorio a Proba Teórica 2 (PT2).

Para aprobar a materia seguindo o sistema de **avaliación continua**, é necesario cumprir dúas condicións: entregar o proxecto realizado nas semanas 10 a 14 do curso, e obter unha nota final (NF) igual ou superior a 5. A nota final (que se calculará como a media harmónica entre a suma das probas prácticas e a suma das probas teóricas) virá dada pola seguinte expresión:

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

Onde:

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

A Proba Final Teórica (PFT) avalía o coñecemento dos contidos introducidos nas sesións maxistras.

A Proba Final Práctica (PFP) avalía o proxecto entregado. De maneira indirecta, tamén avalía o coñecemento dos contidos introducidos nas sesións maxistras e nas prácticas de laboratorio.

Nótese que a aplicación da media armónica implica que non é posible aprobar a asignatura si algunha das notas (NP ou NT) é inferior a 3.3.

A avaliación continua consta das tarefas que se detallan nesta guía e non son recuperables, é dicir, se un alumno non pode cumprilas no prazo estipulado, o profesor non ten a obrigaón de repetirlas. A cualificación obtida nas tarefas avaliábles será válida tan só para o curso académico no que se realicen.

Antes da realización ou entrega de cada tarefa indícarase a data e procedemento de revisión das cualificacións obtidas. O alumnado terá opción a coñecer a cualificación de cada tarefa e revisar a corrección nun prazo aproximado de dúas semanas.

Para aprobar a materia no sistema de **avaliación ao final do cuadrimestre**, será necesario entregar un proxecto como o realizado polos alumnos de avaliación continua nas semanas 10 a 14 do curso, e haberá que obter polo menos unha nota final igual ou superior a 5. Este sistema consistirá nunha proba de resolución de exercicios (parte teórica) e unha proba no laboratorio que avaliará o proxecto (parte práctica). A nota final (que se calculará como a media harmónica entre a nota de práctica e a nota de teoría) virá dada pola seguinte expresión:

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

Onde:

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

A todos os alumnos que concorran ao exame final da materia, calcularánselles ambas as notas: a **nota de avaliación continua** e a **nota de avaliación ao final de cuadrimestre**. A nota final que se lle outorgará será a maior de ambas. A cualificación será de "Non Presentado" nos seguintes casos:

- En caso de optar pola avaliación continua, só si non realiza ningunha proba logo da Proba Práctica 1 (PP1).

- En caso de optar pola avaliación ao final do cuatrimestre, só si non realiza ningunha das probas finais (PFT e PFP).

Nas convocatorias extraordinarias, para aprobar a materia, será necesario entregar un proxecto como o realizado polos alumnos de avaliación continua nas semanas 10 a 14 do curso, e haberá que obter polo menos unha nota final igual ou superior a 5. Nestas convocatorias, realizarase unha proba de resolución de exercicios (parte teórica) e unha proba no laboratorio que avaliará o proxecto (parte práctica). A nota final (que se calculará como a media harmónica entre a nota de práctica e a nota de teoría) virá dada pola seguinte expresión:

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

Onde:

- Se o alumno preséntase á proba de resolución de exercicios, NT será a nota obtida na devandita proba. Se non, NT será a nota de teoría obtida na avaliación ordinaria.
- Se o alumno preséntase á proba no laboratorio, NP será a nota obtida na devandita proba. Se non, NP será a nota de prácticas obtida na avaliación ordinaria.

A cualificación obtida en calquera das tarefas evaluables será válida tan só para o curso académico no que se realicen, é dicir, non se garda ningunha nota dun curso para o seguinte.

En caso de detección de plaxio nalgún dos traballos/probas realizadas a cualificación será de suspenso (0) e os profesores comunicarán á dirección da escola o asunto para que tome as medidas que considere oportunas.

Bibliografía. Fontes de información

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

Osvaldo 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

Recomendacións

Materias que continúan o temario

Programación II/V05G300V01302

Materias que se recomenda ter cursado previamente

Informática: Arquitectura de ordenadores/V05G300V01103

Outros comentarios

A materia Programación II é unha continuación desta materia no segundo curso.