



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 Pazos Arias, José Juan Ramos Cabrer, Manuel Rodríguez Hernández, Pedro Salvador			
Correo-e	pedro.rodriguez@uvigo.es			
Web	http://fatic.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">1. O algoritmo e as súas distintas representacións: diagrama de fluxo, pseudocódigo, linguaxe natural.2. Implementación do algoritmo mediante unha linguaxe de programación. Paradigmas de programación: A programación modular e a programación estruturada.3. A linguaxe C e a función main(). Código fonte e código obxecto. O compilador e o intérprete.4. 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">1. O alfabeto. Derivacións recursivas de secuencias sintácticamente válidas. Identificadores, números. Constantes simbólicas: A directiva #define e macros. Uso do cualificador const.2. 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 *3. O operador sizeof. Operadores aritméticos. O operador de asignación. Conversión automática de tipos e mediante o operador cast.4. 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">1. Avaliación de expresións con operadores relacionais e operadores booleanos.2. Instrucións de selección: switch, if, if anidado. O operador ternario (?:)3. As instrucións iterativas e a súa importancia na programación modular: while, do while e for. Instrucións break e continue.
Tema 4: Variables indexadas	<ol style="list-style-type: none">1. Declaración de tipo de variables indexadas (Arrays). Asignación de memoria para arrays multidimensionais.2. Arrays unidimensionais e punteiros: aritmética de punteiros. Arrays de caracteres: O carácter de fin de cadea.3. Os arrays de lonxitude variable no estándar C99.4. Asignación dinámica de memoria a arrays de 1 e 2 dimensións: as funcións malloc(), calloc(), realloc().
Tema 5: Funcións	<ol style="list-style-type: none">1. Declaración e definición de funcións. Variables locais, estáticas e globais. Valor de retorno dunha función.2. Parámetros actuais e parámetros formais. Paso de parámetros por valor e por referencia: uso de punteiros. Paso de parámetros por liña de comando á función main().3. Elaboración e uso de bibliotecas de funcións. Funcións de biblioteca que xestionan arrays de caracteres.4. Compilación modular. As directivas condicionais nun ficheiro de cabeceira.5. Funcións recursivas: vantaxes e desvantaxes.
Tema 6: Variables tipo struct	<ol style="list-style-type: none">1. Variables tipo struct: declaración global. Campos dun struct. Punteiros a struct. Os operadores . (punto) e -> (frecha).2. O struct e un punteiro a struct como parámetro e valor de retorno dunha función.3. O typedef con declaracións non triviais.4. Estructuras máis complexas: struct anidados, array de struct.5. Xestión dinámica en creación de listas lineais, listas circulares, árbores.6. Inserción e remoción de variables nunha lista
Tema 7: Ficheiros	<ol style="list-style-type: none">1. Ficheiros de texto: funcións fopen(), fclose().2. Diferentes funcións de entrada/saída para ficheiros: fprintf(), fscanff(), fgets(), feof().3. As funcións con acceso directo ao ficheiro.4. Manexo da información entre ficheiros e listas.5. Estructura dos nodos en listas enlazadas de forma simple.6. Paso de ficheiros a lista e viceversa.

Planificación

Horas na aula

Horas fóra da aula

Horas totais

Actividades introductorias	2	0	2
Lección maxistral	24	24	48
Prácticas de laboratorio	12	14	26
Aprendizaxe baseado en proxectos	8	24	32
Probas prácticas, de execución de tarefas reais e/ou simuladas.	5	15	20
Outras	5	15	20
Informe de prácticas	0	2	2

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

Metodoloxía docente

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

Atención personalizada

Metodoloxías	Descrición
Lección maxistral	O profesorado da materia proporcionará atención individual e personalizada ao alumnado durante o curso, solucionando as súas dúbidas e preguntas. As dúbidas atenderanse de forma presencial (durante a propia sesión maxistral, ou durante o horario establecido para as titorías). O horario de titorías establecerase ao principio do curso e publicárase na páxina web da materia
Prácticas de laboratorio	O profesorado da materia proporcionará atención individual e personalizada ao alumnado durante o curso, solucionando as súas dúbidas e preguntas. Así mesmo, o profesorado orientará e guiará ao alumnado durante a realización das tarefas que teñen asignadas nas prácticas de laboratorio. As dúbidas atenderanse de forma presencial (durante as propias prácticas, ou durante o horario establecido para as titorías). O horario de titorías establecerase ao principio do curso e publicárase na páxina web da materia.
Aprendizaxe baseado en proxectos	O profesorado da materia proporcionará atención individual e personalizada ao alumnado durante o curso, solucionando as súas dúbidas e preguntas. Así mesmo, o profesorado orientará e guiará ao alumnado durante a realización das tarefas que teñen asignadas para a realización do proxecto. As dúbidas atenderanse de forma presencial (durante as propias sesións de seguimento do traballo, ou durante o horario establecido para as titorías). O horario de titorías establecerase ao principio do curso e publicárase na páxina web da materia.

Avaliación

	Descrición	Cualificación	Resultados de Formación e Aprendizaxe
Aprendizaxe baseado en proxectos	Desenvolverase un proxecto nas últimas semanas do curso, e entregarase o código C que o implementa. A avaliación do proxecto realízase mediante a proba práctica final .	25	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 práctica final que avaliará a realización das prácticas de laboratorio e do proxecto.	20	B4 C12
Outras	Cada 4 semanas, farase un exame que pode conter: - cuestións de resposta curta - cuestións tipo test - resolución de problemas e/ou exercicios Este exame avaliará o coñecemento dos conceptos introducidos nas sesións maxistras. Farase unha proba teórica 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: Itrrodución de teoría + Temas 1 e 2
- Semana 2: Tema 3 | Introducción de práctica + Práctica 1
- Semana 3: Temas 3 e 4 | Práctica 2
- Semana 4: Tema 4 + **Proba Teórica 1** (PT1) | **Proba 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 + **Proba Teórica 2** (PT2) | **Proba Práctica 2** (PP2)
- Semana 9: Temas 5 e 6 | Práctica 6
- Semana 10: Tema 6 | Finalización de Prácticas + Proxecto (1h)
- Semana 11: Tema 6 | Proxecto (2h) + Entrega do deseño do proxecto (seudocódigo ou diagrama de fluxo)
- Semana 12: Tema 7 + **Proba Teórica 3** (PT3) | Proxecto (1h) - **Proba Práctica 3** (PP3)
- Semana 13: Tema 7 - Proxecto (2h)
- Semana 14: Proxecto (2h)
- Previo ao período de exames, entrega do proxecto: codificación e memoria de documentación
- Período de exames: **Proba Teórica Final** (PTF) | **Proba Práctica Final** (PPF)

Seguindo as directrices propias da titulación, ofrécese ao alumnado que curse 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 a Proba Práctica 2 (PP2).

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

$$NFC = (2*NTC*NPC)/(NTC+NPC)$$

Onde:

- Nota Teórica por Avaliación Continua: $NTC = 0.1*PT1+0.1*PT2+0.2*PT3+0.6*PTF$
- Nota Práctica por Avaliación Continua: $NPC = 0.1*PP1+0.1*PP2+0.2*PP3+0.5*PPF+0.1*PDD$

A Proba Teórica Final (PTF) é un exame que pode conter cuestións de resposta curta e/ou tipo test e/ou de resolución de problemas e/ou exercicios. Avalía o coñecemento dos contidos introducidos nas sesións maxistras.

A Proba Práctica Final (PPF) avalía o proxecto entregado. Aínda que o proxecto desenvólvese en grupo, se evalúa de forma individual. De maneira indirecta, a PPF tamén avalía o coñecemento dos contidos introducidos nas sesións maxistras e nas prácticas de laboratorio.

A **Proba de Deseño e Documentación** (PDD) avalía a calidade do pseudocódigo ou diagrama de fluxo que describe o proxecto (entregado na semana 11), e a memoria de documentación do proxecto entregada antes do período de exames. Nótese que a aplicación da media harmónica implica que non é posible aprobar a materia se algunha das notas (NPC ou NTC) é inferior a 3.3.

A avaliación continua consta das tarefas que se detallan nesta guía e non son recuperables, é dicir, se un/ha alumno/a non pode cumprilas no prazo estipulado, o profesorado non ten a obrigaición de repetirlas.

Antes da realización ou entrega de cada tarefa indicárase 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 (deseño,

codificación e documentación) como o realizado polo alumnado de avaliación continua nas semanas 10 a 14 do curso, e haberá que obter unha nota final igual ou superior a 5. Esta modalidade consistirá nas mesmas probas que a de avaliación continua (aínda que con distinto peso na nota final), é dicir unha proba con cuestións de resposta curta e/ou tipo test e/ou de resolución de problemas e/ou exercicios (Proba Teórica Final, PTF) e unha proba no laboratorio que avaliará o proxecto (Proba Práctica Final, PPF). A nota final por avaliación final (que se calculará como a media harmónica entre a nota de práctica e a nota de teoría) virá dada pola seguinte expresión:

$$NFF = (2*NTF*NPF)/(NTF+NPF)$$

Onde:

- Nota Teórica por Avaliación Final: $NTF = PTF$
- Nota Práctica por Avaliación Final: $NPF = 0.9*PPF+0.1*PDD$

A todo o alumnado que concorra ao exame final da materia, calcularánselle ambas dúas notas: a **nota de avaliación continua** (NFC) e a **nota de avaliación ao final de cuadrimestre (NFF)**. 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 cuadrimestre, só si non realiza ningunha das probas finais (PTF e PPF).

Nas convocatorias extraordinarias, para aprobar a materia, será necesario entregar un proxecto (deseño, codificación e documentación) como o realizado polo alumnado de avaliación continua nas semanas 10 a 14 do curso, e haberá que obter unha nota final igual ou superior a 5. Nestas convocatorias, realizarase unha proba con cuestións de resposta curta e/ou tipo test e/ou de resolución de problemas e/ou exercicios (Proba Teórica Extraordinaria, PTE) e unha proba no laboratorio que avaliará o proxecto (Proba Práctica Extraordinaria, PPE). A nota final por avaliación extraordinaria (que se calculará como a media harmónica entre a nota de práctica e a nota de teoría) virá dada pola seguinte expresión:

$$NFF = (2*NTF*NPF)/(NTF+NPF)$$

Onde:

- Nota Teórica por Avaliación Extraordinaria (NTE): se o/a alumno/a preséntase á Proba Teórica Extraordinaria, NTE será a nota obtida na devandita proba:

$$NTE = PTE$$

Se non, NT será a nota de teoría obtida na avaliación ordinaria.

- Nota Práctica por Avaliación Extraordinaria (NPE): se o/a alumno/a preséntase á Proba Práctica Extraordinaria, NPE será a suma ponderada da nota obtida na devandita proba e a nota obtida na proba de deseño e documentación:

$$NPE = 0.9*PPE+0.1*PDD$$

Se non, NP será a nota de prácticas obtida na avaliación ordinaria.

En todas as convocatorias, en calquera das modalidades de cálculo da nota, no caso de incumprir a condición necesaria de presentar o proxecto, a nota final (calculada segundo a correspondente fórmula) estará coutada superiormente a un máximo de 4.5 puntos.

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 o profesorado comunicará á dirección da escola o asunto para que tome as medidas que considere oportunas.

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

Materias que continúan o temario

Programación II/V05G300V01302

Materias que se recomienda ter cursado previamente

Informática: Arquitectura de ordenadores/V05G300V01103

Outros comentarios

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