



Guía Docente				
Datos Identificativos			2014/15	
Asignatura (*)	Tecnoloxía da Programación	Código	614311202	
Titulación	Enxeñeiro Técnico en Informática de Sistemas			
Descriptorios				
Ciclo	Período	Curso	Tipo	Créditos
1º e 2º Ciclo	1º cuatrimestre	Segundo	Troncal	6
Idioma	Castelán			
Prerrequisitos				
Departamento	Computación			
Coordinación		Correo electrónico		
Profesorado		Correo electrónico		
Web	campusvirtual.udc.es/moodle			
Descrición xeral	<p>A materia de Tecnoloxía da Programación atópase vinculada principalmente ás materias de Programación e Estrutura de Datos e da Información, de primeiro curso, e á materia de Metodoloxía da Programación de segundo curso.</p> <p>Tendo o anterior en conta, presuponse que o alumno adquiriu os seguintes coñecementos:</p> <ul style="list-style-type: none">- Coñecementos básicos da estrutura dun programa, tanto no seu deseño algorítmico, como na súa codificación. Isto inclúe conceptos como variables, tipos de datos, operador, estruturas de control, etc.- Modulación e estruturación, tocando aspectos como procedementos e funcións, variables globais e locais, paso de parámetros por valor e por referencia, módulos e compilación separada, etc.- Estruturas de datos estáticas e dinámicas, incluíndo arrays, rexistros, conxuntos e punteiros.- Algoritmos básicos de clasificación e procura.- Ficheiros: tipos, organización, métodos de acceso e operacións.- Aproximación ao concepto de recursividade no seo dun algoritmo.- Comprender o concepto de tipo abstracto de dato.- Coñecer unha colección de tipos abstractos de datos paradigmáticos. O alumno disporá da capacidade para elixir e manipular distintas estruturas de datos.- Adquirir a habilidade para identificar, deseñar e implementar os tipos abstractos de datos adecuados a unha aplicación concreta. <p>Coa vista posta no futuro profesional, os coñecementos adquiridos nesta materia xogarán un papel relevante no desenvolvemento de aplicacións. O alumno será capaz de desenvolver unha aproximación disciplinada á especificación, implementación, verificación e proba de programas .</p>			

Competencias da titulación	
Código	Competencias da titulación
A3	Interpretar as especificacións funcionais encamiñadas ao desenvolvemento das aplicacións informáticas.
B1	Aprender a aprender.
B2	Resolver problemas de forma efectiva.
B3	Aplicar un pensamento crítico, lóxico e creativo.
B4	Aprendizaxe autónoma.
B5	Traballar de forma colaborativa.
B7	Comunicarse de maneira efectiva en calquera contorno de traballo.
B8	Traballar en equipos de carácter interdisciplinar.
B9	Capacidade para tomar decisións.
B11	Razoamento crítico.
B12	Capacidade para a análise e a síntese.
B13	Capacidade de comunicación.



Resultados da aprendizaxe			
Competencias de materia (Resultados de aprendizaxe)	Competencias da titulación		
1. Capacitar ao alumno para a resolución de problemas de forma eficiente	A3	B2 B11 B12	
2. Adquisición de bos hábitos á hora de programar	A3	B2 B4	
3. Capacidade de aplicar os coñecementos adquiridos á práctica	A3	B2 B3 B9 B11	
4. Capacidade de adaptarse a novas situacións	A3	B1 B3 B9 B11 B12	
5. Traballo en equipo	A3	B5 B7 B8 B13	

Contidos	
Temas	Subtemas
Bloque Temático I. Introducción	-----
1. Cálculo proposicional	1.1. Evaluación de proposicións 1.2. Proposicións como conxuntos de estados 1.3. Leis de equivalencia
2. Predicados	2.1. Extensión do rango de estado 2.2. Cuantificadores 2.3. Identificadores libres e ligados 2.4. Substitución textual
3. Arrays, notación e simplificación de expresións	3.1. Arrays dunha dimensión 3.2. Simplificación de expresións 3.3. Arrays multidimensionales
Bloque Temático II. Cálculo de predicados	-----
4. Uso de asercións na documentación de programas	4.1. Especificación de programas 4.2. Representación de valores iniciais e finais de variables 4.3. Esquemas de proba
5. O transformador de predicados wp	5.1. Definición do transformador de predicados 5.2. Propiedades do wp 5.3. Estratexia de demostración de corrección
6. A linguaxe GCL: Guarded Command Language	6.1. Os comandos skip, abort e composición 6.2. O comando asignación 6.3. A estrutura alternativa 6.4. A estrutura iterativa 6.5. Funcións e procedementos
Bloque Temático III. Verificación e desenvolvemento de programas	-----



7. Desenvolvemento de programas con estruturas alternativas	7.1 Principios básicos 7.2. Exemplos
8. Desenvolvemento de programas con estruturas repetitivas	8.1. Desenvolvendo primeiro o gardián 8.2. Progresando cara á finalización
9. Desenvolvemento de invariantes	9.1. A teoría do globo 9.2. Borrado dunha conxunción 9.3. Substitución dunha constante por unha variable 9.4. Agrandar o rango dunha variable 9.5. Combinar precondición e postcondición
10. Desenvolvemento de funcións cota	10.1. Utilización da notación do problema e a súa solución 10.2. Utilización da orde lexicográfico
Bloque Temático IV: Linguaxe Java e JML	-----
11. Introducción á linguaxe Java	-
12. Introducción á linguaxe JML	-

Planificación			
Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Lecturas	0	100	100
Solución de problemas	0	45	45
Proba obxectiva	3	0	3
Atención personalizada	2	0	2

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

Metodoloxías	
Metodoloxías	Descrición
Lecturas	Lectura y estudio del material bibliográfico de la asignatura.
Solución de problemas	Desarrollo personal de ejercicios de verificación formal de algoritmos.
Proba obxectiva	Proba escrita utilizada para a avaliación da aprendizaxe.

Atención personalizada	
Metodoloxías	Descrición
	Para a súa realización é importante consultar co profesor os avances que se vaian realizando progresivamente para ofrecer as orientacións necesarias en cada caso e asegurar así a calidade dos traballos de acordo aos criterios que se indicarán.

Avaliación		
Metodoloxías	Descrición	Cualificación
Proba obxectiva	Exámen final que se cualificará cunha nota de 0 a 10.	100
Outros		

Observacións avaliación
A los alumnos que tengan evaluada la práctica de programación de JAVA y JML, en los dos cursos académicos anteriores al actual, la prueba objetiva final supondrá el 90% de la nota final y la nota de dicha práctica el 10% restante. En el caso de no tener dicha práctica entregada, la prueba objetiva final se corresponderá con el 100% de la nota.

Fontes de información



Bibliografía básica	<ul style="list-style-type: none">- Peña Marí, R. (2005). Diseño de programas: Formalismo y abstracción. Madrid. Prentice-Hall- Arnold, K.; Gosling, J.; Holmes, D. (2001). El lenguaje de programación Java. Madrid. Addison-Wesley- Arnow, D.; Weiss, G. (2000). Introducción a la programación en Java. Un enfoque orientado a objetos. Madrid. Addison-Wesley- Balcazar, J.L. (1993). Programación metódica. McGraw-Hill- (2007). The Java Modeling Language (JML). http://www.cs.iastate.edu/leavens/JML- Gries, D. (1981). The Science of Programming. Springer-Verlag
Bibliografía complementaria	<ul style="list-style-type: none">- P. Naughton (1996). Manual de Java. Osborne McGraw-Hill- Backhouse, R.C. (1986). Program construction and verification. Prentice-Hall- Kaldewaij, A. (1990). Programming: The derivation of algorithms. Prentice-Hall- Apt, K.R., Olderog, E. (1997). Verification of sequential and concurrent programs. Springer-Verlag

Recomendacións

Materias que se recomenda ter cursado previamente

Materias que se recomenda cursar simultaneamente

Programación Orientada a Obxectos/614311636

Materias que continúan o temario

Estrutura de Datos e da Información/614311102

Matemática Discreta/614311107

Programación/614311109

Observacións

(*A Guía docente é o documento onde se visualiza a proposta académica da UDC. Este documento é público e non se pode modificar, salvo casos excepcionais baixo a revisión do órgano competente dacordo coa normativa vixente que establece o proceso de elaboración de guías