



| Guía Docente | | | | |
|-----------------------|---|--------------------|-----------|----------|
| Datos Identificativos | | | | 2015/16 |
| Asignatura (*) | Tecnoloxía da Programación | Código | 614111202 | |
| Titulación | Enxeñeiro en Informática | | | |
| Descritores | | | | |
| Ciclo | Período | Curso | Tipo | Créditos |
| 1º e 2º Ciclo | 2º cuatrimestre | Segundo | Troncal | 6 |
| Idioma | CastelánGalego | | | |
| Modalidade docente | Presencial | | | |
| Prerrequisitos | | | | |
| Departamento | Computación | | | |
| Coordinación | | Correo electrónico | | |
| Profesorado | | Correo electrónico | | |
| Web | http://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. Tendo o anterior en conta, presuponse que o alumno adquiriu os seguintes coñecementos: - 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. 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 / Resultados do título | |
|-------------------------------------|--|
| Código | Competencias / Resultados do título |
| A3 | Concibir e planificar o desenvolvemento de aplicacións informáticas complexas ou con requisitos especiais. |
| 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. |
| B11 | Razoamento crítico. |
| B12 | Capacidade para a análise e a síntese. |
| B13 | Capacidade de comunicación. |

| Resultados da aprendizaxe | | | |
|--|----|------------------|-------------------------------------|
| Resultados de aprendizaxe | | | Competencias / Resultados do título |
| 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 | | B1 B2 B4 | |
| 4. Capacidade de adaptarse a novas situacións | | B1 B3 B11 B12 | |
| 5. Traballo en equipo | | B5 B7 B8 B13 | |

| Contidos | |
|--|---|
| Temas | Subtemas |
| 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 |
| 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. Corrección parcial | 6.1. Asignacións 6.2. Arrays 6.3. If 6.4. While |
| 7. Corrección total | 7.1. While |
| 8. Alternativas | 8.1. Postcondición máis forte (sp) |
| 9. Ferramentas e técnicas de proba na práctica | 9.1 Sistemas de control de versións. Subversión 9.2 Sistemas de compilación Ant 9.3 Ferramentas de probas de unidade JUnit 9.4 Calculo automático de métricas de cobertura 9.5 Técnicas de probas no Desenvolvemento Software |

| Planificación | | | | |
|-----------------------|---------------------------|---|-------------------------|--------------|
| Metodoloxías / probas | Competencias / Resultados | Horas lectivas (presenciais e virtuais) | Horas traballo autónomo | Horas totais |
| Proba obxectiva | | 3 | 0 | 3 |
| Actividades iniciais | | 146 | 0 | 146 |



| | | | | |
|--|--|---|---|---|
| Atención personalizada | | 1 | 0 | 1 |
| *Os datos que aparecen na táboa de planificación son de carácter orientativo, considerando a heteroxeneidade do alumnado | | | | |

| Metodoloxías | |
|----------------------|---|
| Metodoloxías | Descrición |
| Proba obxectiva | Proba escrita |
| Actividades iniciais | Temario e máis exercicios vistos no curso 2010/2011 |

| Atención personalizada | |
|------------------------|-----------------------|
| Metodoloxías | Descrición |
| Proba obxectiva | Aclaración de dúbidas |

| Avaliación | | | |
|-----------------|---------------------------|---|---------------|
| Metodoloxías | Competencias / Resultados | Descrición | Cualificación |
| Proba obxectiva | | <p>Teoría:</p> <ul style="list-style-type: none"> - Proba escrita da parte de teoría. - Cualificación mínima: 4 de 10. - 70% da cualificación global. - A cualificación da proba escrita se garda entre convocatorias se é un 5 ou máis. <p>Existe tamén a posibilidade de facer de novo o exame para mellorar a cualificación (con isto, o alumno renunciaría á anterior cualificación).</p> <p>Práctica:</p> <ul style="list-style-type: none"> - Proba escrita da parte de práctica. - Cualificación mínima: 4 de 10. - 30% da cualificación global. - A cualificación da proba escrita se garda entre convocatorias se é un 5 ou máis. <p>Existe tamén a posibilidade de facer de novo o exame para mellorar a cualificación (con isto, o alumno renunciaría á anterior cualificación).</p> | 100 |
| Outros | | | |

| Observacións avaliación |
|--|
| Cualificación global mínima: 5 de 10. |
| No caso de non superar a teoría ou a práctica, a nota publicada será como máximo 4,5 |

| Fontes de información | |
|------------------------------------|---|
| Bibliografía básica | <ul style="list-style-type: none"> - Daniel Bolaños et al. (). Pruebas de Software y JUnit. Un análisis en profundidad y ejemplos prácticos. - Gries, David (). The Science of Programming. |
| Bibliografía complementaria | <ul style="list-style-type: none"> - (). http://junit.sourceforge.net/. - (). http://svnbook.red-bean.com/. - Winskel, Glynn (). The Formal Semantics of Programming Languages. |

| Recomendacións |
|--|
| Materias que se recomenda ter cursado previamente |
| |
| Materias que se recomenda cursar simultaneamente |



| Materias que continúan o temario |
|---|
| Estrutura de Datos e da Información/614111102 |
| Matemática Discreta I/614111107 |
| Programación/614111109 |
| Metodoloxía da Programación/614111205 |
| Programación Orientada a Obxectos/614111636 |
| 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