



Guía Docente				
Datos Identificativos			2014/15	
Asignatura (*)	Programación Orientada a Objetos	Código	614111636	
Titulación				
Descriptorios				
Ciclo	Período	Curso	Tipo	Créditos
1º e 2º Ciclo	1º cuatrimestre	Todos	Optativa	4
Idioma	Castelán			
Prerrequisitos				
Departamento	Computación			
Coordinación	Alonso Ríos, David	Correo electrónico	david.alonso@udc.es	
Profesorado	Alonso Ríos, David	Correo electrónico	david.alonso@udc.es	
Web				
Descrición xeral	<p>La programación orientada a objetos (POO) es, hoy en día, el paradigma de programación dominante en el desarrollo de sistemas informáticos. La POO surge como un paso más en la evolución de la programación imperativa añadiendo nuevas propiedades (como herencia, polimorfismo, etc.) a los tipos abstractos de datos. El carácter optativo de la asignatura puede llevar a engaño ya que la relativa antigüedad de los planes de estudio (1994) y de las troncalidades (1991) no recogen correctamente la situación actual en el ámbito de la programación (Java sólo tiene poco más de 10 años de antigüedad).</p> <p>La filosofía de aprendizaje marcada en el plan de estudios se configura como ?Imperativo primero?, es decir, primero se explican los lenguajes imperativos para dar paso más adelante a la programación orientada a objetos. De esta forma el aprendizaje de la programación se hace más gradual, aunque el paso a los conceptos de la orientación a objetos obliga a un cambio de filosofía en la forma de programar cuya adaptación puede resultar compleja al alumno.</p> <p>Las competencias académicas que se pretenden desarrollar son importantes para cursar otras materias ligadas directa o indirectamente con la programación. Entre las más directamente relacionadas podemos destacar: (II, ITIG e ITIS) Proyecto fin de Carrera, (II) Análisis de Sistemas Informáticos, Diseño de Sistemas Informáticos, Integración de Sistemas, Análisis y Diseño Orientado a Objetos, e (ITIG) Principios de Análisis Informático.</p> <p>Esta materia también resulta muy interesante a la hora de configurar habilidades en el contexto del ámbito profesional ya que el paradigma de la orientación a objetos es el dominante dentro de los lenguajes de programación más utilizados profesionalmente (Java, C#, C++, VisualBasic, Delphi, etc.).</p>			

Competencias da titulación	
Código	Competencias da titulación

Resultados da aprendizaxe			
Competencias de materia (Resultados de aprendizaxe)	Competencias da titulación		
Conocer las diferencias entre el paradigma de programación imperativa tradicional y el paradigma de la orientación a objetos.	A1	B1	C3
Comprender los conceptos básicos en los que se basa la orientación a objetos: clases, objetos, mensajes, etc.	A1	B1 B2 B4	C3
Analizar, comprender y relacionar las propiedades básicas propias de la orientación a objetos: herencia, polimorfismo, ligadura dinámica, etc.	A1	B1 B2 B3 B4	C3



Conocer los aspectos básicos del diseño UML, sobre todo lo que involucra el diseño estático (diagrama de clases) y dinámico (diagrama de secuencia) de un programa orientado a objetos.	A1 A3	B1 B2 B3 B4	C3
Conocer, comprender e identificar los principios y patrones de diseño básicos para lograr un programa orientado a objetos flexible y reusable.	A1 A3	B1 B2 B3 B4	C3
Construir clases básicas que representen objetos del dominio y que encapsulen el estado y el comportamiento de dichos objetos.	A1	B2 B3 B4 B9 B10 B12	C3
Desarrollar y organizar clases sacando provecho de las propiedades propias de la orientación a objetos para desarrollar código flexible y reusable.	A1 A3	B2 B3 B4 B9 B10 B12	C3
Diseñar un programa orientado a objetos utilizando el estándar UML y desarrollar posteriormente el código resultante de dicho diseño.	A1 A3 A7	B2 B3 B4 B5 B9 B10 B12	C3
Diseñar un programa utilizando principios y patrones de diseño garantizando así que las propiedades propias de la orientación a objetos se utilizan adecuadamente.	A1 A3	B2 B3 B4 B5 B9 B10 B12	C3
Usar la programación orientada a objetos para resolver problemas reales.	A1	B2 B3 B4 B9 B10 B12	C3 C6
Manejar con solvencia un entorno integrado de desarrollo (IDE) en la programación	A1 A6	B2 B3 B4	C3
Asumir la responsabilidad del trabajo y las acciones propias.		B5 B6	C4
Valorar y respetar el trabajo propio y el ajeno.		B5 B6	C4
Colaborar con los compañeros para llevar a cabo una práctica en grupo.		B5 B7	C4



Valorar la importancia del análisis y el diseño en el desarrollo de programas orientados a objetos.		B10	C3
		B15	C6
Preferir las buenas prácticas de programación orientada a objetos sobre los antipatrones (malas prácticas) propias de los principiantes o de los malos programadores.		B15	C6

Contidos	
Temas	Subtemas
EXAMEN DE TEORÍA	TEMARIO DE LOS ÚLTIMOS AÑOS DE LA ASIGNATURA (resumido a continuación). Para más información consultar el Moodle o hablar con el profesor.
Introducción	Paradigmas de la programación Programación orientada a objetos El lenguaje Java
Modelado visual de objetos: UML	Introducción Elementos básicos del UML Diseño estático: diagrama de clases Diseño dinámico: diagrama de secuencia
Elementos básicos de la orientación a objetos	Clases Objetos
Propiedades básicas de la orientación a objetos	Abstracción Encapsulamiento Modularidad Jerarquía Polimorfismo Tipificación Ligadura dinámica
Patrones de diseño	Introducción Patrones creacionales: inmutable, singleton Patrones de comportamiento: estrategia, estado, iterador, observador, método plantilla Patrones estructurales: composición, adaptador
Principios de diseño	Principio abierto-cerrado Principio de sustitución de Liskov Diseño por contrato y principio de subcontratación Principio de inversión de la dependencia Principios de diseño y herencia
Herramientas	NetBeans JUnit
EXAMEN DE PRÁCTICAS	TEMARIO: SELECCIÓN DE CONCEPTOS CLAVE DE POO RELACIONADOS CON LA APLICACIÓN PRÁCTICA DEL TEMARIO DE TEORÍA Y CON LOS EJERCICIOS PRÁCTICOS PLANTEADOS EN LOS ÚLTIMOS AÑOS DE LA ASIGNATURA. Para más información consultar el Moodle o hablar con el profesor.

Planificación			
Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Proba obxectiva	5	90	95
Atención personalizada	5	0	5

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



## Metodoloxías

Metodoloxías	Descrición
Proba obxectiva	Exame escrito, dividido en: <ul style="list-style-type: none"><li>- Exame de teoría.</li><li>- Exame de prácticas.</li></ul> Observacións: En proceso de extinción non hai ningún tipo de avaliación continua. Non se recollerán exercicios prácticos, sólo haberá exames. Para máis información consultar o Moodle ou falar co profesor.

## Atención personalizada

Metodoloxías	Descrición
Proba obxectiva	Resolución de dúbidas

## Avaliación

Metodoloxías	Descrición	Cualificación
Proba obxectiva	Exame de teoría: <ul style="list-style-type: none"><li>- Cualificación mínima exixida: 4 de 10.</li><li>- 60% da cualificación global.</li><li>- A cualificación desta proba escrita se garda entre convocatorias se é un 5 ou máis. Existe tamén a posibilidade de facer de novo o exame para mellorar a cualificación (con isto, o alumno renunciaría á anterior cualificación).</li></ul> Exame de prácticas: <ul style="list-style-type: none"><li>- Proba escrita da parte de práctica.</li><li>- Cualificación mínima exixida: 4 de 10.</li><li>- 40% da cualificación global.</li><li>- A cualificación desta proba escrita se garda entre convocatorias se é un 5 ou máis. Existe tamén a posibilidade de facer de novo o exame para mellorar a cualificación (con isto, o alumno renunciaría á anterior cualificación).</li></ul>	100
Outros		

## Observacións avaliación

Cualificación en actas:- Aprobado ou superior: Cualificación final maior ou igual que 5 e cumprir todos os requerimentos exixidos para superar a materia.- Suspenso: Cualificación final menor que 5 ou cualificación final maior o igual que 5 e non cumprir todos os requerimentos exixidos para superar a materia. No segundo caso, a cualificación en actas será 4,5.- Non Presentado: Equivale a non presentado a ambos exames finais (teoría e prácticas). Para máis información consultar o Moodle ou falar co profesor.
--

## Fontes de información

Bibliografía básica	
Bibliografía complementaria	

## Recomendacións

Materias que se recomenda ter cursado previamente
---



Análise de Sistemas Informáticos/614111402

Deseño de Sistemas de Información/614111403

Integración de Sistemas/614111503

Proxecto fin de Carreira/614111505

Análise e Deseño Orientado a Obxectos/614111602

**Materias que se recomenda cursar simultaneamente**

Metodoloxía da Programación/614111205

**Materias que continúan o temario**

Estrutura de Datos e da Información/614111102

Programación/614111109

**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