Guía Docente				
Datos Identificativos				2012/13
Asignatura (*)	Programación Orientada a Obxectos		Código	614111636
Titulación	Enxeñeiro en Informática			
Descriptores				
Ciclo	Período	Curso	Tipo	Créditos
1º e 2º Ciclo	1º cuadrimestre	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	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). 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. 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. 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.).			

Competencias da titulación	
Código	Competencias da titulación
A1	Aprender de maneira autónoma novos coñecementos e técnicas avanzadas axeitadas para a investigación, o deseño e o desenvolvemento de sistemas e servizos informáticos.
A3	Concibir e planificar o desenvolvemento de aplicacións informáticas complexas ou con requisitos especiais.
A6	Avaliar, definir, seleccionar e auditar plataformas hardware e software para a execución e desenvolvemento de aplicacións e servizos informáticos.
A7	Saber especificar, deseñar e implementar un sistema de información, empregando bases de datos.
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.
B6	Comportarse con ética e responsabilidade social como cidadán e como profesional.
B7	Comunicarse de maneira efectiva en calquera contorno de traballo.
B9	Capacidade para tomar decisións.
B10	Capacidade de xestión da informática (captación e análises da información).

B12	Capacidade para a análise e a síntese.
B15	Motivación pola calidade.
C3	Utilizar as ferramentas básicas das tecnoloxías da información e as comunicacións (TIC) necesarias para o exercicio da súa profesión e para a aprendizaxe ao longo da súa vida.
C4	Desenvolverse para o exercicio dunha cidadanía aberta, culta, crítica, comprometida, democrática e solidaria, capaz de analizar a realidade, diagnosticar problemas, formular e implantar solucións baseadas no coñecemento e orientadas ao ben común.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.

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 B15	C3 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**
Introducción	Paradigmas de la programación Programación orientada a objetos El lenguaje Java
Elementos básicos de la orientación a objetos	Clases Objetos
Propiedades básicas de la orientación a objetos	Abstracción Encapsulamiento Modularidad Jerarquía Polimorfismo Tipificación Ligadura dinámica
Modelado visual de objetos: UML	Introducción Elementos básicos del UML Diseño estático: diagrama de clases Diseño dinámico: diagrama de secuencia
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

Planificación			
Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Sesión maxistral	25	25	50
Solución de problemas	2	2	4
Prácticas de laboratorio	2	12	14
Traballos tutelados	4	20	24
Proba de resposta múltiple	2	0	2
Foro virtual	0	2	2
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
Sesión maxistral	En las clases presenciales de teoría, el profesor realizará una breve descripción de los contenidos temáticos y de los objetivos básicos perseguidos, con el fin de dotar al alumno de una visión global de la materia. Además se tratará de establecer interrelaciones con otros conceptos previamente adquiridos, de forma que se pueda establecer una línea temporal, y expondrá la bibliografía recomendada. Seguidamente pasará a desarrollar los contenidos teóricos utilizando como método la clase magistral.
Solución de problemas	Dentro de las clases presenciales también se incluirán clases o ejercicios de resolución de problemas que serán resueltos ?in situ? con la colaboración activa de los alumnos. Estas clases o ejercicios se plantean como una forma de afianzar los conceptos teóricos explicados por el profesor y tratan de fomentar la participación de los alumnos, el diálogo abierto y la valoración de soluciones.
Prácticas de laboratorio	Las clases de prácticas consistirán principalmente en el desarrollo de una práctica final de la asignatura en la que se intentará ejercitar todos los contenidos expuestos en las clases teóricas. El enunciado de la práctica detallará el problema y las especificaciones que el alumno deberá respetar estrictamente. Posteriormente, la labor del profesor consistirá en la resolución de errores, solución de dudas, corrección de malos hábitos, etc. durante las horas de tutoría. El desarrollo de la práctica comenzará con el diseño estático y dinámico del programa en base al estándar UML y terminará con su implementación en el lenguaje Java mediante el entorno de desarrollo NetBeans. Publicación del enunciado: Se indicará durante el curso. Fecha límite de entrega: Se indicará durante el curso. Dada la carga de trabajo, se recomienda trabajar en grupo. Los grupos serían de 4 personas como máximo. Cualquier problema con los grupos deberá ser comunicado inmediatamente a los profesores.

Traballos tutelados	Se planteará la realización de un boletín de ejercicios (repartido en varios boletines pequeños) que los alumnos deberán resolver en parejas dentro o fuera del horario de prácticas. Los alumnos también podrán consultar dudas con los profesores de prácticas durante las horas de tutorías. Los ejercicios se implementarán en el lenguaje Java mediante el entorno de desarrollo NetBeans. Publicación del enunciado: Varias fechas, se indicarán durante el curso. Fecha límite de entrega: Varias fechas, se indicarán durante el curso. Dada la carga de trabajo, se recomienda trabajar en grupo. Los grupos serían de 2 personas como máximo. Cualquier problema con los grupos deberá ser comunicado inmediatamente a los profesores.
Proba de resposta múltiple	Examen teórico de preguntas de respuesta múltiple sobre la materia impartida a lo largo del curso
Foro virtual	El Campus Virtual de la UDC permite tener foros en los que los alumnos y los profesores debatiran distintos aspectos relacionados con la asignatura

Atención personalizada	
Metodoloxías	**Descrición**
Foro virtual Prácticas de laboratorio Traballos tutelados	Resolución de dudas y cuestiones que puedan surgir durante el desarrollo de las prácticas en particular y de la asignatura en general

Avaliación		
Metodoloxías	**Descrición**	**Cualificación**
Prácticas de laboratorio	PRÁCTICA DE DISEÑO: Porcentaje de la nota global: 20%. Requisito para superar la asignatura: Sacar al menos un 4 de 10. Criterios de evaluación: La nota tendrá en cuenta no sólo el código desarrollado sino también la calidad de la memoria entregada, la calidad del análisis y diseño realizado, la calidad de los modelos UML desarrollados, etc. Más detalles en las instrucciones de la propia práctica. Plagios: Copiar en algo supone un 0 en todo. Para todas las personas del grupo que plagie y para para todas las personas del grupo que se deje plagiar. Convocatorias extraordinarias (septiembre, diciembre): - Se guarda la nota si es mayor o igual que 4 de 10. - Se permite intentar subir nota, y ello significa renunciar a la nota antigua.	20

Traballos tutelados	BOLETÍN DE EJERCICIOS: Porcentaje de la nota global: 20%. Requisito para superar la asignatura: no. Criterios de evaluación: En la convocatoria ordinaria la evaluación de los ejercicios será siguiendo los criterios estadísticos ECTS (quién haga más ejercicios correctos más nota tendrá). Más detalles en las instrucciones de los propios boletines. Plagios: Copiar en algo supone un 0 en todo. Para todas las personas del grupo que plagie y para para todas las personas del grupo que se deje plagiar. Convocatorias extraordinarias (septiembre, diciembre): - Se guarda la nota si es mayor o igual que 5 de 10. - Se permite intentar subir nota, y ello significa renunciar a la nota antigua. - Los enunciados originales sólo son vigentes durante la convocatoria ordinaria. En las convocatorias extraordinarias se sustituirán por otros nuevos.	20
Proba de resposta múltiple	PRUEBAS TEÓRICAS: Porcentaje de la nota global: 60%. Repartida entre examen final (50%) y controles de evaluación continua (10%) en convocatoria ordinaria. Requisito para superar la asignatura: Sacar al menos un 4 de 10 en el examen final. Criterios de evaluación: En un examen tipo test las respuestas incorrectas restan puntos para compensar el efecto del azar. Por ejemplo, si hay cuatro alternativas posibles, tres errores anulan un acierto. Más detalles en el propio enunciado. Plagios: Copiar en algo supone un 0 en todo. Para quien plagie y para quien se deje plagiar. Convocatorias extraordinarias (septiembre, diciembre): - Se guarda la nota si es mayor o igual que 5 de 10. - Se permite intentar subir nota, y ello significa renunciar a la nota antigua. - La evaluación continua sólo es vigente durante la convocatoria ordinaria. En las convocatorias extraordinarias habrá sólo un examen final, que supondrá el 60% de la nota global.	60
Outros		

Observacións avaliación

DERECHO A EVALUACIÓN:

El alumno tendrá derecho a realizar todas las actividades evaluables en las fechas indicadas.

El alumno que, por un motivo debidamente justificado, aún no se encontrase oficialmente matriculado, tendría los mismos derechos a realizar las actividades evaluables que los demás alumnos. El alumno no tendría privilegios para realizar nada fuera de plazo.

Se rechazaría cualquier circunstancia que diese lugar a agravios comparativos.

PUBLICACIÓN DE NOTAS:

Las calificaciones definitivas del examen final se publicarán después de la revisión, antes del cierre de actas.

Las calificaciones definitivas de las actividades evaluables previas al examen final se publicarán al menos siete días antes de la fecha del examen.

CALIFICACIÓN EN ACTAS:

- Aprobado o superior: Nota final mayor o igual que 5 y haber cumplido todos los requisitos para superar la asignatura.
- Suspenso: Nota final menor que 5 o nota final mayor o igual que 5 y no haber cumplido todos los requisito para superar la asignatura. En el segundo caso, la calificación en actas será 4,5.
- No Presentado: Equivale a no presentado al examen final.

ALUMNOS REPETIDORES:

Los alumnos repetidores pueden tomar una de las siguientes decisiones (no es necesario comunicársela al profesor):

1) Decidir desde el principio presentarse a la convocatoria de diciembre y no participar en la evaluación continua del nuevo curso para evitar trabajo innecesario.

2) Decidir desde el principio presentarse a la convocatoria de diciembre y, como repaso o como precaución, participar en la evaluación continua del nuevo curso. El repetidor tendría los mismos derechos que los alumnos de primera matrícula excepto el de trabajar en grupo.

3) Decidir desde el principio no presentarse a la convocatoria de diciembre y participar en la evaluación continua del nuevo curso igual que los alumnos de primera matrícula. Este sería el único caso en el que se les permitiese a los repetidores trabajar en grupo. Se recuerda a los alumnos que tienen la obligación moral de ser honestos con sus compañeros.

Si el repetidor aprueba en la convocatoria de diciembre, no se le computará ninguna nota de evaluación continua del nuevo curso.

Si el repetidor suspende o no se presenta en la convocatoria de diciembre, se abrirá a continuación un breve plazo de entrega para que tenga la posibilidad de entregar los boletines del nuevo curso fuera de plazo. A cada cosa entregada fuera de plazo se le descontarán 2 puntos sobre 10 de la nota.

IMPLANTACIÓN DEL GRADO PARA EL PRÓXIMO CURSO:

El curso siguiente, la asignatura de Programación Orientada a Objetos pasará a encontrarse en extinción. Una vez empezado el proceso de extinción, no habrá ningún tipo de docencia ni evaluación continua de la asignatura de Programación Orientada a Objetos. Sólo habrá pruebas finales que corresponderán a los contenidos teóricos y prácticos descritos. Se informará de todo en el Moodle.

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

El estudio de la asignatura no debe plantearse como una actividad de estudio memorístico de los temas presentados en clase, sino que deberá tener una orientación más práctica. Por ejemplo, se debe saber qué es la ligadura dinámica, pero también se debe saber poner un ejemplo en Java de la misma o reconocer su aparición en un código Java.

Por otro lado hay que tener en cuenta que una asignatura como la programación orientada a objetos incluye un gran número de nuevos términos o conceptos (polimorfismo, genericidad, patrones, etc.) que no sólo hay que saber aplicar en la práctica sino que también hay que conocer en qué consisten, cuáles son sus ventajas e inconvenientes, cuándo es recomendable utilizarlos. Todos estos aspectos deben ser objeto de estudio por parte del alumno, si bien se intentará evitar en el examen final preguntas que prioricen el aprendizaje memorístico.

En cuanto a los boletines de ejercicios y a la práctica final se recomienda encarecidamente al alumno que no haga uso de actividades académicas deshonestas como pueden ser la copia o préstamo de ejercicios o prácticas ya que pueden llevar al suspenso automático de la asignatura (tanto del que ha copiado como del que ha permitido que le copien).

Las recomendaciones para mejorar o recuperar un examen suspenso son las siguientes:

* Ir a la revisión de exámenes y conocer en qué se ha fallado
* Ir a tutorías y preguntar las dudas que se tengan al profesor, no ir con dudas significativas al examen. Sois alumnos por lo que la única pregunta estúpida es aquella que no se hace.
* Intentar mejorar la nota de prácticas mejorando aquellos aspectos en los que se ha tenido menor puntuación
* Realizar más ejercicios de los boletines intentando relacionar la resolución de los ejercicios con los aspectos teóricos vistos en clase.

(*)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