



Guía Docente				
Datos Identificativos				2023/24
Asignatura (*)	Deseño Software	Código	614G01015	
Titulación	Grao en Enxeñaría Informática			
Descritores				
Ciclo	Período	Curso	Tipo	Créditos
Grao	1º cuatrimestre	Segundo	Obrigatoria	6
Idioma	CastelánInglés			
Modalidade docente	Presencial			
Prerrequisitos				
Departamento	Ciencias da Computación e Tecnoloxías da InformaciónComputación			
Coordinación	Mosqueira Rey, Eduardo	Correo electrónico	eduardo.mosqueira@udc.es	
Profesorado	Alonso Ríos, David Blanco Mallo, Eva Mosqueira Rey, Eduardo Paz López, Alejandro Paz Ruza, Jorge Pérez Sánchez, Beatriz Rodríguez Arias, Alejandro	Correo electrónico	david.alonso@udc.es eva.blanco@udc.es eduardo.mosqueira@udc.es alejandro.paz.lopez@udc.es j.ruza@udc.es beatriz.perezs@udc.es alejanro.rodriguez.arias@udc.es	
Web				
Descrición xeral	<p>O Deseño Software é unha fase chave dentro do ciclo de vida do software que establece a ligazón entre os requisitos dun sistema e a súa posterior implementación. O deseño máis habitual hoxe en día é o deseño baseado na orientación a obxectos, que consiste en desenvolver un programa en base a obxectos que intercambian mensaxes.</p> <p>Esta materia introducirá ao alumnado nos elementos e propiedades básicas da orientación a obxectos usando unha linguaxe orientada a obxectos como Java. Aprenderase tamén a como reflectir os artefactos propios do deseño nunha linguaxe de modelado coma a Linguaxe Unificada de Modelado (UML).</p> <p>Finalmente presentaranse aqueles principios básicos que representan un bo deseño e aprenderase a identificar aqueles problemas típicos de deseño e as súas solucións máis comúns representadas como patróns de deseño.</p>			

Competencias / Resultados do título	
Código	Competencias / Resultados do título
A7	Capacidade para deseñar, desenvolver, seleccionar e avaliar aplicacións e sistemas informáticos que aseguren a súa fiabilidade, seguranza e calidade, conforme a principios éticos e á lexislación e normativa vixente.
A13	Coñecemento, deseño e utilización de forma eficiente dos tipos e estruturas de datos máis adecuados á resolución dun problema.
A14	Capacidade para analizar, deseñar, construír e manter aplicacións de forma robusta, segura e eficiente, elixindo o paradigma e as linguaxes de programación máis adecuados.
B1	Capacidade de resolución de problemas
B2	Traballo en equipo
B3	Capacidade de análise e síntese
B4	Capacidade para organizar e planificar
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.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.

Resultados da aprendizaxe	
Resultados de aprendizaxe	Competencias / Resultados do título



Identificar o deseño software como unha das fases do ciclo de vida do software	A7 A13 A14	B3 B4	C3
Coñecer os principios e propiedades básicas da orientación a obxectos	A7 A13 A14	B1 B2 B3 B4	C3 C6
Plasmar un deseño software utilizando os artefactos propios dunha linguaxe de modelado coma UML	A7 A13 A14	B1 B2 B3 B4	C3 C6
Coñecer os principios básicos que representan un bo deseño software	A7 A13 A14	B1 B2 B3 B4	C3 C6
Identificar problemas típicos de deseño e as súas solucións máis comúns	A7 A13 A14	B1 B2 B3 B4	C3 C6
Usar un deseño como guía para a implementación do software	A7 A13 A14	B1 B2 B3 B4	C3 C6
Aprender unha linguaxe orientada a obxectos e aspectos relacionados (IDE, probas, repositorios, etc.)	A13	B1 B2 B3 B4	C3 C6

Contidos	
Temas	Subtemas
1. Introducción	? Deseño software ? Analise e deseño orientado a obxectos
2. Elementos Básicos da Orientación a Obxectos	? Clases e obxectos ? Identidade de obxectos ? Estado de obxectos ? Comportamento de obxectos
3. Características Básicas da Orientación a Obxectos	? Abstracción e encapsulamento ? Modularidade ? Xerarquía ? Polimorfismo ? Tipificación ? Ligadura Dinámica
4. Linguaxe Unificada de Modelado (UML)	? Introducción ? Bloques básicos do UML ? Deseño estático: Diagramas de clases ? Deseño dinámico: Diagramas de interacción ? Outros diagramas
5. Principios de Deseño	? Calidade no deseño ? Principios SOLID ? Tipos de herdanza



6. Patróns de Deseño	? Introducción aos patróns de deseño ? Patróns elementais ? Deseños adaptables aos cambios ? Patróns e coleccións de obxectos ? Deseños débilmente acoplados ? Outros patróns e principios
Prácticas	? Introducción a Java ? Programación en parella ? Probas de software ? Repositorios de código

Planificación				
Metodoloxías / probas	Competencias / Resultados	Horas lectivas (presenciais e virtuais)	Horas traballo autónomo	Horas totais
Sesión maxistral	A7 A13 A14 B1 B3 C6	30	45	75
Prácticas de laboratorio	A7 A13 A14 B1 B2 B3 B4 C3 C6	20	30	50
Seminario	A7 A13 A14 B1 B2 B3 B4 C3 C6	10	10	20
Proba obxectiva	A7 A13 A14 B1 B3 C6	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
Sesión maxistral	Clases expositivas de presentación dos coñecementos teóricos empregando diferentes recursos: pizarra, proxección de material en formato electrónico, apuntes en formato electrónico e os recursos facilitados polo profesorado da asignatura no Campus Virtual da UDC.
Prácticas de laboratorio	Prácticas baseadas nos coñecementos que cada estudante vai adquirindo nas clases teóricas. Estes traballos serán desenvolvidos preferiblemente en grupo. Empregarase unha ferramenta de modelado para construír os artefactos de deseño e aplicarase unha linguaxe orientada a obxectos (Java) para realizar a implementación dos mesmos.
Seminario	Seminarios nos que se realizarán actividades relacionadas cos coñecementos prácticos fundamentalmente.
Proba obxectiva	Proba escrita mediante a que se valoran os coñecementos adquiridos polo estudantado. Cada estudante deberá aplicar tanto os seus coñecementos tanto a nivel teórico coma a nivel práctico.

Atención personalizada	
Metodoloxías	Descrición
Prácticas de laboratorio Seminario	<p>A atención personalizada ao estudantado comprende non só as titorías, presenciais ou virtuais, para a discusión de dúbidas, senón tamén as seguintes actuacións:</p> <ul style="list-style-type: none"> <li>- Seguemento do labor realizado nas prácticas de laboratorio propostas polo profesorado.</li> <li>- Avaliación dos resultados obtidos nas prácticas, participación en seminarios realizados por cada estudante.</li> <li>- Encontros personalizados para resolver dúbidas sobre os contidos da asignatura.</li> </ul>



Avaliación			
Metodoloxías	Competencias / Resultados	Descrición	Cualificación
Prácticas de laboratorio	A7 A13 A14 B1 B2 B3 B4 C3 C6	Realización de exercicios prácticos baseados na programación en Java, na orientación a obxectos, o deseño de probas, a linguaxe de modelado UML e o uso de principios e patróns de deseño.  Se se detecta algún exercicio copiado nunha práctica, esta será anulada na súa totalidade (cualificación cero), tanto o orixinal como a copia.  É obrigatorio obter unha nota mínima de 4 sobre 10 para poder facer media cos outros elementos avaliados.	33
Seminario	A7 A13 A14 B1 B2 B3 B4 C3 C6	Os seminarios son sesións de carácter práctico dirixidas polo profesorado nas que se comentan aspectos útiles relacionados coas prácticas.  Os seminarios non inclúen a entrega de traballos por parte do alumnado, polo que non é unha actividade evaluable.	0
Proba obxectiva	A7 A13 A14 B1 B3 C6	Proba escrita realizada ao final do curso sobre contidos teórico-prácticos.  A proba obxectiva é obrigatoria para aprobar a materia e tamén é obrigatorio obter unha nota mínima de 4 sobre 10 para poder facer media cos outros elementos avaliados.	67

### Observacións avaliación

En caso de non chegar á nota mínima de 4 na proba obxectiva e nas prácticas de laboratorio, en calquera das oportunidades, implicará que non se poida obter máis dun 4,5 na nota final da materia.

Considerarase "presentado" á asignatura:

Os que se presenten ao exame da proba obxectiva na 1ª oportunidade. Os que se presenten ao exame da proba obxectiva da 2ª oportunidade ou presenten a práctica da 2ª oportunidade. Aspectos a ter en conta para a avaliación de segunda oportunidade (xullo):

Aspectos xenerais (2ª Op.):

As porcentaxes son as mesmas que as da 1ª Oportunidade. Tamén rexe a norma de obter un mínimo dun 4 na proba obxectiva e nas prácticas de laboratorio. Se te presentas a algunha parte na 2ª oportunidade (proba obxectiva ou prácticas) anulas a nota da primeira nesa parte. A nota da proba obxectiva e das prácticas de laboratorio da primeira oportunidade gardarse para a segunda oportunidade se son iguais ou superior a 4. Prácticas de laboratorio (2ª Op.):

Establecerase un prazo para presentar unha práctica na 2ª oportunidade. Aspectos a ter en conta en caso de matrícula a tempo parcial:

Elimínase a obrigatoriedade de asistir as actividades nas que se poida esixir presencialidade, salvo na proba obxectiva.

### Fontes de información

<b>Bibliografía básica</b>	<ul style="list-style-type: none"><li>- Sierra, K., Bates, B. (2005). Head First Java. O'Reilly</li><li>- Schildt, H. (2018). Java 9. Anaya Multimedia</li><li>- Booch J.; Rumbaugh J. y Jacobson I. (2006). El Lenguaje Unificado de Modelado (2ª ed.) The Unified Modeling Language (2nd ed.). Addison Wesley</li><li>- Martin, R.C. (2004). UML para programadores Java. UML for Java Programmers. Pearson</li><li>- Gamma, E.; Helm, R.; Johnson, R. y Vlissides J. (1996). Patrones de Diseño : Elementos de Software Orientado a Objetos Reutilizable. Design Patterns: Elements of Reusable Object-oriented Software.. Addison Wesley</li></ul>
----------------------------	--



<b>Bibliografía complementaria</b>	<ul style="list-style-type: none"><li>- Schildt, H. (2019). Java: The Complete Reference. McGraw-Hill Education</li><li>- Urma, R.G. (2014). Java 8 in Action. Manning</li><li>- Rumbaugh, J.; Jacobson, I. y Booch, J. (2004). El Lenguaje Unificado de Modelado: Manual de Referencia. The Unified Modeling Language: Reference Manual. Addison Wesley</li><li>- Bloch, J. (2017). Effective Java (3rd ed.). Addison Wesley</li><li>- Martin, R.C. (2012). Código limpio : manual de estilo para el desarrollo ágil de software. Clean Code: A Handbook of Agile Software Craftsmanship. Anaya Multimedia</li><li>- Larman C. (2005). Applying UML and Patterns, 3rd ed.. Prentice-Hall</li><li>- Freeman, E., Freeman, E., Bates, B. (2004). Head First Design Patterns. O'Reilly</li></ul>
------------------------------------	--

## Recomendacións

### Materias que se recomenda ter cursado previamente

Programación I/614G01001

Programación II/614G01006

### Materias que se recomenda cursar simultaneamente

Paradigmas de Programación/614G01014

### Materias que continúan o temario

Proceso Software/614G01019

Interfaces Persoa Máquina/614G01022

Internet e sistemas distribuidos/614G01023

## Observacións

A materia asume que o alumnado sabe programar e coñece as estruturas de datos (Programación II) aínda que nunca utilizaron unha linguaxe orientada a obxectos. Ao principio do curso, e segundo vanse introducindo os conceptos propios da orientación a obxectos, o alumnado familiarízase cos aspectos básicos da linguaxe de programación Java.

(\*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