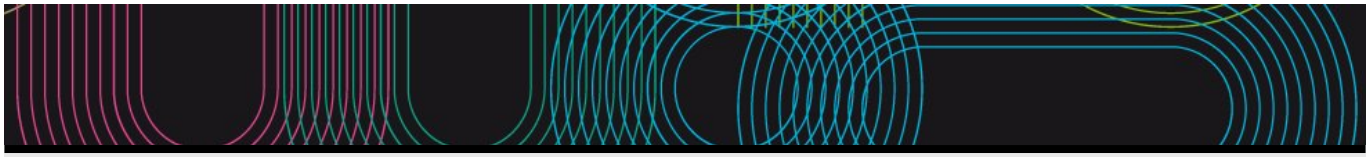

DATOS IDENTIFICATIVOS				
Programación II				
Materia	Programación II			
Código	V05G300V01302			
Titulación	Grao en Enxeñaría de Tecnoloxías de Telecomunicación			
Descritores	Creditos ECTS	Sinale	Curso	Cuadrimestre
	6	OB	2	1c
Lingua de impartición	Castelán			
Departamento	Enxeñaría telemática			
Coordinador/a	Blanco Fernández, Yolanda			
Profesorado	Blanco Fernández, Yolanda Fernández Masaguer, Francisco			
Correo-e	yolanda@det.uvigo.es			
Web	http://www.faitic.es			
Descrición xeral	O obxectivo xeral da materia é proporcionarle ao alumnado os fundamentos teóricos e as competencias prácticas que lle permitan analizar, deseñar, desenvolver e depurar aplicacións informáticas seguindo o paradigma orientado a obxectos. Esta é unha materia eminentemente práctica e neste sentido está orientada ao traballo do alumnado na realización dun ou varios proxectos. Para facilitar o desenvolvemento dos proxectos, na materia, realizarase primeiramente unha moi breve introdución á disciplina de Enxeñaría do Software, conectándoa co paradigma da programación orientada a obxectos (POO) e limitándoa só ás etapas de análise, deseño, implementación e depuración. A continuación analizaranse en detalle os elementos da POO, utilizando elementos e diagramas UML que utilizará o alumnado nos seus desenvolvementos. Para alcanzar este obxectivo xeral os contidos que se tratarán na materia pódense resumir nos seguintes ítems: - Conceptos básicos de Enxañaría do Software. - Conceptos básicos da orientación a obxectos: clases e obxectos. - Encapsulación. Principio de ocultación. Conceptos de *desacoplamiento e cohesión - Herdanza, abstracción, polimorfismo e reutilización - Relacións entre clases: xeneralización, asociación e dependencia. - Comunicación entre obxectos: métodos, eventos, mensaxes. - Persistencia. Almacenamento en ficheiros e en bases de datos. - Xeración, captura e procesamento de excepcións. - Linguaxe de modelado UML.			

Competencias	
Código	
B6	CG6 Facilidade para o manexo de especificacións, regulamentos e normas de obrigado cumprimento.
B14	CG14 Capacidade para utilizar ferramentas informáticas de procura de recursos bibliográficos ou de información.
C50	(CE50/T18) Capacidade de desenvolver, interpretar e depurar programas utilizando os conceptos básicos da Programación Orientada a Obxectos (POO): clases e obxectos, encapsulación, relacións entre clases e obxectos, e herdanza.

C51	(CE51/T19) Capacidade de a aplicación básica das fases de análises, deseño, implantación e depuración de programas na POO.
C52	(CE52/T20) Capacidade de manexo de ferramentas CASE (editores, depuradores).
C53	(CE53/T21) Capacidade de desenvolvemento de programas atendendo aos principios básicos de calidade da enxeñaría do software, tendo en conta as principais fontes existentes en normas, estándares e especificacións.

Resultados de aprendizaxe

Resultados previstos na materia	Resultados de Formación e Aprendizaxe	
Comprender os aspectos básicos da Programación Orientada a Obxectos (POO).	B14	C50
Coñecer os principais diagramas UML para a documentación nas fases de análise e deseño de programas de acordo á POO.	B6 B14	C52 C53
Desenvolver habilidades no proceso de análise, deseño, implementación e depuración de aplicacións de acordo á POO, tendo en conta os estándares principais e normas de calidade.	B6 B14	C51 C53
Adquirir unha madurez básica en técnicas de desenvolvemento e depuración de programas para permitir a aprendizaxe autónoma de novas capacidades e linguaxes de programación.	B6	C51 C52 C53

Contidos

Tema	
1. Introdución ao paradigma orientado a obxectos	a. Breve introdución á materia e á súa organización b. Nacemento do paradigma c. Bases: clases e obxectos d. Conceptos de encapsulación, herdanza (xeneralización), e polimorfismo e. Breve introdución a UML
2. Encapsulación	a. Clases, interfaces e paquetes b. Métodos e variables membro. Visibilidade. Resolución de ámbito. c. Método constructor d. Paso de parámetros: punteiros e referencias e. Punteiros a obxectos
3. Herdanza	a. Clases derivadas e tipos de herdanza b. Clases abstractas c. Herdanza múltiple d. Clase object
4. Deseño orientado a obxectos	a. Fundamentos de deseño b. Conceptos básicos da Enxeñaría do Software c. Utilización de diagramas UML
5. Polimorfismo	a. Sobrecarga e sobreescritura b. Clases abstractas e interfaces c. Clases xenéricas
6. Xestión de excepcións	a. Fundamentos de excepcións b. Manipulación de excepcións en Java

Planificación

	Horas na aula	Horas fóra da aula	Horas totais
Lección maxistral	28	42	70
Resolución de problemas	5	8	13
Resolución de problemas de forma autónoma	6	17	23
Estudo de casos	3	9	12
Aprendizaxe baseado en proxectos	7	18	25
Estudo de casos	1	0	1
Resolución de problemas	3	0	3
Práctica de laboratorio	2	0	2
Proxecto	1	0	1

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

Metodoloxía docente

	Descrición
Lección maxistral	Clases que combinarán a exposición dos conceptos tratados na materia coa realización de pequenos exercicios. O docente ou o propio alumnado resolveraos individualmente ou en grupo. O obxectivo é fomentar o debate na clase e reforzar a adquisición de destrezas. Esta metodoloxía está orientada á adquisición das competencias CE50, CE51 e CE53.

Resolución de problemas	No laboratorio o profesor amosará anacos de código Java para axudar ao alumnado a entender mellor os principais conceptos da POO. Esta metodoloxía está orientada á adquisición das competencias CE50, CE51 and CE53.
Resolución de problemas de forma autónoma	O alumnado resolverá de forma autónoma as prácticas que o profesor propoña no laboratorio. As solucións e as dúbidas que xurdan abordando estes problemas serán discutidas para identificar os erros máis comunmente cometidos. Esta metodoloxía está orientada á adquisición das competencias CE50, CE51, CE53, CG6 e CG14.
Estudo de casos	O profesor supervisará e guiará ós alumnos durante o deseño dos diagramas UML, coa intención de identificar os erros máis comúns nesta fase do proxecto. Esta metodoloxía está orientada á adquisición das competencias CE51 e CE52.
Aprendizaxe baseado en proxectos	Os alumnos implementarán o sistema software proposto polo profesor. Disporán para isto da segunda parte do curso, combinando traballo presencial no laboratorio supervisado polo profesor con traballo non presencial. Esta metodoloxía está orientada á adquisición das competencias CE50, CE53, CG6 e CG14.

Atención personalizada

Metodoloxías	Descrición
Resolución de problemas	Revisión e comentarios de exercicios resoltos. Glosario de erros cometidos con frecuencia. Recomendacións de estilo e organización de código.
Aprendizaxe baseado en proxectos	O profesor supervisará o nivel de entendemento dos alumnos, asistíndoos en dúbidas particulares, posibles erros de deseño e melloras no nivel de código Java.
Resolución de problemas de forma autónoma	Revisión e comentarios con cada grupo das diversas prácticas propostas durante a súa realización. Resolución de erros de compilación e execución. Detección e solución de erros conceptuais.
Estudo de casos	Revisión e crítica xeral do deseño UML de cada grupo durante a súa realización.

Avaliación

	Descrición	Cualificación	Resultados de Formación e Aprendizaxe	
Aprendizaxe baseado en proxectos	O proxecto consiste no deseño final (diagramas UML), o código Java e a documentación correspondente. O código necesariamente ten que compilar e poder ser executado nos ordenadores do laboratorio. Este proxecto pode ser levado a cabo individualmente ou en grupos de dúas persoas, segundo o mecanismo de avaliación elixido por cada estudante. O profesor entrevistará ao alumnado para asegurar a autoría do código entregado e para executar diferente probas de funcionalidade. No caso de grupos, os dous membros teñen que asistir á entrevista. O estudantado contestará as preguntas do profesor individualmente para corroborar así o seu nivel de entendemento e de implicación durante o desenvolvemento do proxecto. Cada estudante ten que identificar a parte do proxecto software que implementou. Os estudantes que, a xuízo do profesor, non consigan demostrar a autoría do código suspenderán a materia na primeira oportunidade. Noutro caso, a nota de cada alumno dependerá de (i) as súas respostas durante a entrevista, (ii) a cantidade de probas de funcionalidade superadas, e (iii) a calidade do código Java no tocante á adopción das técnicas da POO.	35	B6 B14	C50 C53
Estudo de casos	Os estudantes deseñarán un proxecto software utilizando a linguaxe UML, incluindo diferentes tipos de diagramas xunto coa documentación corresponden. O deseño UML pode desenvolverse individualmente ou en grupos de dous alumnos segundo o mecanismo de avaliación elexido. No caso de grupos, a nota de cada membro da parella dependerá da calidade dos diagramas UML propostos como solución.	5		C51 C52
Resolución de problemas	Cada estudante realizará, individualmente e sen material de apoio, un exame na data oficial aprobada pola Xunta de Escola. Este exame combinará problemas, cuestións de resposta curta, elección múltiple e probas de verdadeiro/falso, orientadas a avaliar o nivel de entendemento dos estudantes en relación aos conceptos teóricos expostos nas sesións teóricas da materia.	50		C50 C51 C53
Práctica de laboratorio	Esta proba consiste nun conxunto de prácticas de iniciación Java que axudarán ós alumnos a familiarizarse cunha linguaxe de programación orientada a obxectos. O alumnado que siga a avaliación continua entregará estas prácticas. Serán desenvolvidas en grupos de dous alumnos e a súa nota individual dependerá dos seguintes factores: (i) as respostas de cada alumno ás cuestións propostas polo profesor nunha entrevista, e (ii) a calidade e o correcto funcionamento do código Java entregado.	10		C50 C51 C52 C53

Outros comentarios sobre a Avaliación

Existen dous mecanismos de avaliación: avaliación continua (AC) e avaliación única (AU), que elixirá o alumnado considerando as seguintes condicións:

- A AC inclúe as 4 probas descritas na sección anterior deste documento (exame de teoría, prácticas de iniciación Java, deseño UML e implementación dun proxecto software).
- Os alumnos que opten por AU deben implementar o proxecto software (publicado en faiTIC) individualmente.
- Mediante a entrega do deseño UML do proxecto, os estudantes comprométense a ser avaliados mediante AC, renunciando así ao mecanismo de AU. En virtude do devandito compromiso, estes alumnos non poderán figurar como "Non presentados".
- Os estudantes que non entreguen o deseño UML renuncian ao mecanismo de AC, sendo necesariamente avaliados mediante AU. Non será posible unirse á AC nas seguintes probas.
- A planificación das diferentes probas de avaliación intermedia aprobarase nunha Comisión Académica de Grado (CAG) e estará dispoñible ao principio do cuadrimestre.
- As probas de AC só se levarán a cabo nas datas estipuladas polos profesores. Non poderán repetirse máis tarde.
- As notas de AC e doutros exames e proxectos prácticos non serán válidas máis aló do ano académico actual.
- En caso de detección de copia en calquera das probas (probas curtas, exames parciais ou exame final), a cualificación final será e SUSPENSO (0) e o feito comunicaráselle á dirección do Centro para os efectos oportunos.

Procedemento de avaliación en primeira oportunidade para alumnos que opten por AC:

Parte teórica: Exame (50%). Exame individual sen ningún tipo de material de apoio. É a terceira proba descrita na sección de avaliación. A nota deeste exame só poderá ser gardada para a segunda oportunidade en caso de obter 4.5 ou máis puntos (sobre 5).

Parte práctica. Inclúense as seguintes probas:

- **Prácticas de iniciación Java** (10%). Desenvolveranse en grupos de 2 alumnos. É a cuarta proba descrita na sección de avaliación.
- **Proxecto** (40%). Desenvolverase en grupo de dous alumnos. O proxecto consta de dúas partes:
    - **Deseño UML** (5%). É a segunda proba descrita na sección de avaliación.
    - **Implementación Java** (35%). É a primeira proba descrita na sección de avaliación, incluído o código Java, a documentación Javadoc correspondente e a entrevista de autoría co profesor.

Para superar a materia mediante o mecanismo de AC os alumnos deben reunir os seguintes requisitos:

- Conseguir polo menos 1/3 da nota máxima da parte teórica.
- Conseguir polo menos 1/3 da nota máxima da implementación Java do proxecto da parte práctica.
- Conseguir unha nota final (parte teórica + parte práctica) igual ou superior a 5 puntos (sobre 10).
- Se a nota final é igual ou maior ca 5 pero o alumno non alcanza as notas mínimas mencionadas anteriormente, a súa nota final será suspenso (4.5).

Procedemento de avaliación na primeira oportunidade para alumnos que opten por AU:

Parte teórica: **Exame** (50%). Exame individual sen ningún tipo de material de apoio. É a terceira proba descrita na sección de avaliación. Os alumnos que obteñan en primeira oportunidade unha nota nesta parte de 4.5 puntos ou máis (sobre 5), poderán rescatar a súa cualificación sen necesidade de repetir o exame de teoría.

Parte práctica: **Proxecto** (50%). Debe desenvolverse individualmente. É a primeira proba descrita no apartado de avaliación, incluindo o deseño UML, o código Java e a correspondente documentación Javadoc, así como a entrevista de autoría co profesor.

Para superar a materia mediante o mecanismo de AU os alumnos deben reunir os seguintes requisitos:

- Conseguir polo menos 1/3 da nota máxima da parte teórica.
- Conseguir polo menos 1/3 da nota máxima da parte práctica.

- Conseguir unha nota final (parte teórica + parte práctica) igual ou superior a 5 puntos (sobre 10).
- Se a nota final é igual ou maior ca 5 pero o alumno non alcanza as notas mínimas mencionadas anteriormente, a súa nota final será suspenso (4.5).

Procedemento de avaliación na segunda oportunidade:

Parte teórica: **Exame** (50%). Exame individual sen ningún tipo de material de apoio. É a terceira proba descrita na sección de avaliación. A nota de este exame non se gardará nunca.

Parte práctica: Proxecto (50%). Na avaliación desta parte téñense en conta tres posibles escenarios, segundo o mecanismo de avaliación elixido polo alumno (AC ou AU) e das notas obtidas na parte práctica durante a primera oportunidade. En todos os escenarios, cómpre salientar que os alumnos que optasen por AU na primeira oportunidade deberán realizar o proxecto de forma individual na segunda oportunidade.

[Scenario #1] A nota do proxecto pode gardarse para a segunda oportunidade. Aínda que os alumnos poden rescatar as notas obtidas na primeira oportunidade (só nos supostos descritos máis abaixo), teñen tamén a posibilidade de mellorar a súa calificación entregando unha versión do proxecto requirido na segunda oportunidade (cuxas especificacións publicaranse en faiTIC). Neste caso, os estudantes deberán proporcionar un documento no que se describan os cambios que fagan no deseño da primeira versión do proxecto para poder acomodar as novas funcionalidades esixidas.

Este escenario aplícase a estudantes que optaron por AC na primeira oportunidade, sempre que a nota do seu proxecto sexa igual ou maior ca 1.5 (sobre 4 puntos) e a nota do deseño UML sexa igual ou maior ca 0.3 (sobre 0.5). No caso de que o alumno non rescate a súa nota da primeira oportunidade e opte por entregar as funcionalidades esixidas na segunda oportunidade, a implementación do proxecto avaliarase con ata 3.5 puntos (sobre 10) e o deseño UML con ata 0.5 puntos (sobre 10). A nota das prácticas de iniciación Java (ata 1 punto sobre 10) será a obtida na primeira oportunidade.

Este escenario tamén é válido para os alumnos que optasen por AU na primeira oportunidade, tendo obtido unha nota no proxecto igual ou superior a 2.5 puntos (sobre 5). No caso de optar por unha nova entrega na segunda oportunidade, o proxecto será avaliado con ata 5 puntos (sobre 10).

[Scenario #2] Consérvanse as notas obtidas no deseño UML e nas prácticas de iniciación Java, pero é preciso entregar un novo proxecto na segunda oportunidade que será avaliado sobre 3.5 puntos.

Este escenario é válido para estudantes que opten por AC na primeira oportunidade e obteñan unha nota no proxecto igual ou maior ca 1 punto pero menor ca 1.5 puntos (sobre 4), e unha nota no deseño UML maior ca 0.3 puntos (sobre 0.5).

[Scenario #3] Non é posible rescatar ningunha nota da primeira oportunidade. Neste caso o alumno deberá entrega un novo proxecto que será avaliado con ata 5 puntos (sobre 10).

Este escenario aplícase a estudantes que optaron por AC na primeira oportunidade cunha nota no proxecto menor ca 1 punto (sobre 4) e cunha calificación no deseño UML menor ca 0.3 (sobre 0.5).

Este escenario tamén é válido para estudantes que non entregaron o proxecto durante a primeira oportunidade, e para alumnos que non conseguiron superar a entrevista de autoría realizada co profesor.

Para superar a materia na segunda oportunidade os alumnos deben reunir os seguintes requisitos:

- Conseguir polo menos 1/3 da nota máxima da parte teórica.
- Conseguir polo menos 1/3 da nota máxima da parte práctica (3.5 puntos en AC e 5 puntos en AU).
- Conseguir unha nota final (parte teórica + parte práctica) igual ou superior a 5 puntos (sobre 10).
- Se a nota final é igual ou maior ca 5 pero o alumno non alcanza as notas mínimas mencionadas anteriormente, a súa nota final será suspenso (4.5).

Procedemento de avaliación na convocatoria extraordinaria (fin de carreira):

Parte teórica: **Exame** (50%). Exame individual sen ningún tipo de material de apoio. É a terceira proba descrita na sección de avaliación. A nota desta parte non poderá ser rescatada dun exame de teoría previo en ningún suposto.

Parte práctica: **Proxecto** (50%). Deberá desenvolverse individualmente. É a primeira proba descrita no apartado de avaliación, na que se inclúen o deseño UML, o código Java e a correspondente documentación Javadoc, ademais da entrevista de autoría co profesor.

Para superar a materia na convocatoria extraordinaria (fin de carreira) os alumnos deben reunir os seguintes requisitos:

- Conseguir polo menos 1/3 da nota máxima da parte teórica.
- Conseguir polo menos 1/3 da nota máxima da da parte práctica.
- Conseguir unha nota final (parte teórica + parte práctica) igual ou superior a 5 puntos (sobre 10).
- Se a nota final é igual ou maior ca 5 pero o alumno non alcanza as notas mínimas mencionadas anteriormente, a súa nota final será suspenso (4.5).

Bibliografía. Fontes de información

Bibliografía Básica

W. Savitch, **Absolute Java**, 4ª edición, Pearson, 2010

Y. D. Liang, **Introduction to Java programming**, 8ª, Pearson, 2010

P. Deitel, H. Deitel, **Java: How to program**, 9ª, Pearson, 2011

Bibliografía Complementaria

B. Eckel, **Thinking in Java**, 4ª edición, Prentice-Hall, 2006

P. Niemeyer, D. Leuck, **Learning Java**, 4ª edición, O'Reilly., 2013

Oracle, **Java SE. Oracle**,

Oracle, **Java API Specifications**, 2016

G. Booch, J. Rumbaugh, I. Jacobson, **The Unified Modeling Language User Guide**, 2, Addison-Wesley., 2005

S. Zakhour, S. Hommel, J. Royal, I. Rabinovitch, T. Risser, M. Hoeber, **The Java Tutorial. A short course on the basics**, 4ª edición, Prentice-Hall, 2006

A. Eberhart, S. Fischer, **Java Tools**, Wiley, 2002

M. Page-Jones, **Fundamentals of object-oriented design in UML**, Addison-Wesley, 2002

M. Fowler, **UML Distilled: A Brief Guide to the Standard Object Modeling Language**, 3ª edicion, Addison-Wesley., 2003

Jean-Michel DOUDOUX, **Développons en Java 2.10**, 2016

Recomendacións

Materias que se recomenda ter cursado previamente

Programación I/V05G300V01205