

DATOS IDENTIFICATIVOS				
Programación II				
Materia	Programación II			
Código	V05G306V01110			
Titulación	Grao en Enxeñaría de Tecnoloxías de Telecomunicación - Docencia en inglés			
Descritores	Creditos ECTS	Sinale	Curso	Cuadrimestre
	6	OB	1	2c
Lingua de impartición	#EnglishFriendly Inglés			
Departamento	Enxeñaría telemática			
Coordinador/a	Fernández Iglesias, Manuel José Blanco Fernández, Yolanda			
Profesorado	Blanco Fernández, Yolanda Costa Montenegro, Enrique Fernández Iglesias, Manuel José			
Correo-e	yolanda@det.uvigo.es manolo@uvigo.es			
Web	http://www.faitic.es			
Descrición xeral	O obxectivo xeral da materia é proporcionarlle 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. Materia do programa English Friendly. Os/ as estudantes internacionais poderán solicitar ao profesorado: a) materiais e referencias bibliografías para o seguimento da materia en inglés, b) atender as titorías en inglés, c) probas e avaliacións en inglés.			

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

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
Comprender os aspectos básicos da Programación Orientada a Obxectos (POO).	B14	C50

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
4. Deseño orientado a obxectos	a. Fundamentos de deseño b. Conceptos básicos da Enxeñaría do Software c. Utilización de diagramas UML
3. Herdanza	a. Clases derivadas e tipos de herdanza b. Clases abstractas c. Herdanza múltiple d. Clase object
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	25	35	60
Prácticas de laboratorio	8	18	26
Estudo de casos	3	6	9
Prácticas en aulas informáticas	16	35	51
Resolución de problemas e/ou exercicios	1	0	1
Resolución de problemas e/ou exercicios	2	0	2
Resolución de problemas e/ou exercicios	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 explicación dos conceptos da POO e a resolución de exercicios para a súa aplicación. Tales exercicios poderán ser resoltos polo profesor ou polos alumnos, individualmente ou en grupo. O obxectivo é fomentar o debate na aula e reforzar a adquisición das competencias. Esta metodoloxía está orientada á adquisición das competencias CE50, CE51 e CE53.
Prácticas de laboratorio	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.
Prácticas en aulas informáticas	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
Lección maxistral	O profesor atenderá as dúbidas dos estudantes sobre os conceptos descritos durante as clases maxistrais.
Prácticas de laboratorio	O profesor supervisará o nivel de entendemento dos alumnos, asistíndoos en dúbidas particulares, posibles erros de deseño e melloras no nivel de código Java.
Prácticas en aulas informáticas	Revisión e comentarios durante o desarrollo das prácticas, axudando aos alumnos en tarefas de compilación e execución de programas, además de detectar e correxir erros de concepto.
Estudo de casos	Análise, detección de erros e discusión de posibles melloras nos deseños UML presentados polos estudantes.

Avaliación

	Descrición	Cualificación	Resultados de Formación e Aprendizaxe	
Prácticas de laboratorio	Esta proba consiste nun conxunto de prácticas de iniciación Java que axudarán aos alumnos a familiarizarse cunha linguaxe de programación orientado a obxectos. As prácticas serán entregadas unicamente polos alumnos que sigan a avaliación continua, organizados por parellas, optando a unha cualificación máxima de 1 punto (sobre 5). Dita nota asignarase en función da calidade e correcto funcionamento do código Java entregado.	10		C50 C51 C52 C53
Estudo de casos	Os estudantes deseñarán o proxecto software exposto polo profesor mediante a linguaxe UML, incluíndo os diagramas solicitados e a documentación necesaria para entender as decisións de deseño tomadas. Esta proba será realizada unicamente polos alumnos que opten por avaliación continua. A cualificación máxima é 0.5 puntos (sobre 5) e dependerá da calidade e rigor das decisións de deseño e diagramas expostos.	5		C50 C52
Prácticas en aulas informáticas	O proxecto consiste no deseño final (diagramas UML), o código Java e a documentación Javadoc correspondente. O código necesariamente ten que compilar e poder ser executado nos computadores do laboratorio. A nota asignada a esta proba (entre 0 e 3.5 puntos sobre 5 en AC, e entre 0 e 5 puntos en AU) dependerá do resultado que obteña o alumno nun exame práctico. A cualificación repartirase da seguinte forma: 3 puntos (implementación) + 0.5 puntos (deseño UML final) en AC, e 4 puntos (implementación) + 1 punto (deseño) en AU.	35	B6 B14	C50 C53
Resolución de problemas e/ou exercicios	Cada estudiante realizará individualmente e sen ningún tipo de material de apoio un exame de teoría a metade do cuadrimestre (a data exacta aprobarase na CAG e publicarase na páxina da Escola) sobre os contidos que se expliquen ata a semana anterior á proba. Esta proba só será realizada polos alumnos que opten por avaliación continua, podendo obter unha cualificación máxima de 2 puntos (sobre 5).	20		C50 C51 C53
Resolución de problemas e/ou exercicios	Cada estudiante realizará individualmente e sen ningún tipo de material de apoio un exame de teoría ao termo do cuadrimestre (na data oficial aprobada pola CAG) sobre a totalidade dos contidos presentados na materia. A nota máxima desta proba será 3 puntos (sobre 5) para os alumnos que opten por avaliación continua, e 5 puntos para os que elixan o mecanismo de avaliación única.	30		C50 C51 C53
Resolución de problemas e/ou exercicios	Cada alumno deberá realizar un exame práctico no laboratorio docente no que se pedirá unha modificación menor da implementación do proxecto. A cualificación de devandito exame será apto ou non apto. So se avaliarán os proxectos dos alumnos que superen o exame práctico (a calificación do resto será 0 puntos).	0		C50 C51 C53

Outros comentarios sobre a Avaliación

Existen dous mecanismos de avaliación, continua (AC) e única (AU), que deberán ser elixidos polos alumnos considerando as seguintes condicións:

- A AC inclúe as probas descritas no punto anterior: dous puntuables de teoría, prácticas de iniciación Java, deseño UML, implementación dun proxecto e exame práctico.
- Mediante a entrega do deseño UML os alumnos comprométense a ser avaliados por AC, renunciando así ao mecanismo de AU. En virtude de 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.

- Os alumnos que opten por AU deberán entregar o proxecto individualmente. O resto traballarán por parellas durante toda a parte práctica da asignatura.
- O calendario de tódalas probas de avaliación será aprobado pola CAG e posto a disposición do alumnado ao principio do cuadrimestre de impartición da asignatura.
- As probas de AC só se levarán a cabo nas datas fixadas polos profesores, non podendo repetirse máis tarde.
- As notas de AC e doutros exames e proxectos prácticos só serán válidas ao rematar o ano académico actual.
- En caso de plaxio, o estudante recibirá a nota *Suspenso (0)* e este feito será notificado á Dirección do Centro aos efectos oportunos.

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

- **Parte teórica (50%):** A nota desta parte resulta de sumar as cualificacións dos dous puntuables de teoría descritos anteriormente (a metade e a final de cuadrimestre), con cualificacións máximas de 2 e 3 puntos, respectivamente. Esíxese alcanzar 2 puntos (sobre 5) nesta parte para poder superar a materia. En caso de non aprobar en primera oportunidade, os alumnos que logren esa puntuación poderán conservar a nota para a segunda oportunidade.
- **Parte práctica (50%):** A nota desta parte resulta de sumar as cualificacións obtidas nas prácticas de iniciación Java (ata 1 punto), o deseño UML (ata 0.5 puntos), o exame práctico (apto/non apto) e a implementación do proxecto (ata 3.5 puntos). Esíxese superar o exame práctico (apto) e lograr polo menos 1.5 puntos no proxecto (sobre os 3 puntos relativos á implementación e documentación) para aprobar a materia. En caso de suspender en primeira oportunidade, os alumnos que cumplan as dúas condicións poderán conservar a nota da parte práctica para segunda oportunidade.

Requisitos para aprobar a materia:

1. Conseguir polo menos 2 puntos (sobre 5) na parte teórica.
2. Superar o exame práctico (apto) e conseguir polo menos 1.5 puntos (sobre 3) na implementación/documentació do proxecto proposto na parte práctica.
3. Alcanzar unha nota final, resultado de sumar as cualificacións da parte teórica e práctica, maior ou igual a 5 puntos.
4. Si a nota final é igual ou maior que 5 puntos pero o alumno non alcanza as cualificacións mínimas esixidas na teoría e/ou prácticas, a súa nota final será suspenso (4.5).

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

- **Parte teórica (50%):** A nota desta parte corresponde a un exame de teoría que se celebrará na data oficial aprobada pola CAG. Non se permite ningún tipo de material de apoio. Esíxese alcanzar 2 puntos nesta parte para poder superar a materia. En caso de non aprobar en primeira oportunidade, os alumnos que acaden esa puntuación mínima poderán conservar a cualificación para a segunda oportunidade.
- **Parte práctica (50%):** A nota desta parte corresponde a un proxecto no que se incluirá o deseño UML (1 punto), así coma a implementación Java e a documentación Javadoc corresponden (4 puntos). Ademais, o alumno deberá realizar un exame práctico que será puntuado coma apto ou non apto. Esíxese superar dita proba e lograr polo menos 2 puntos na implementación do proxecto (sobre 4) para aprobar a materia. En caso de suspender en primeira oportunidade, os alumnos que cumplan as dúas condicións poderán conservar a nota da parte práctica para a segunda oportunidade.

Requisitos para aprobar a materia:

1. Conseguir polo menos 2 puntos (sobre 5) na parte teórica.
2. Superar o exame práctico (apto) e conseguir polo menos 2 puntos (sobre 4) na implementación/documentación do proxecto proposto na parte práctica.
3. Alcanzar unha nota final, resultado de sumar as cualificacións da parte teórica e práctica, maior ou igual a 5 puntos.
4. Si a nota final é igual ou maior que 5 puntos pero o alumno non alcanza as cualificacións mínimas fixadas na teoría e/ou prácticas, a súa nota final será suspenso (4.5).

Procedemento de avaliación en segunda oportunidade e convocatoria extraordinaria (fin de carreira):

En segunda oportunidade e na convocatoria extraordinaria non existe a modalidade de AC e o mecanismo de avaliación segue as pautas fixadas no apartado de AU: exame teórico (ata 5 puntos, nota mínima de 2 puntos) + proxecto e exame práctico (ata 5 puntos e apto/non apto, de xeito que para aprobar é preciso superar o exame práctico e obter unha nota mínima de 2 puntos sobre a parte de implementación/documentación). Os alumnos que suspenderan a asignatura previamente (dentro do curso académico actual) poden conservar as notas obtidas anteriormente (sempre que superen a cualificación mínima esixida na parte que desexan conservar) ou ben volver a ser avaliados (neste caso a nota final será sempre a acadada na nova avaliación).

Bibliografía. Fontes de información

Bibliografía Básica

Yolanda Blanco Fernández, **Introducción a Programación Orientada a Objetos**, 1ª edición, Andavira, 2019

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/V05G301V01105