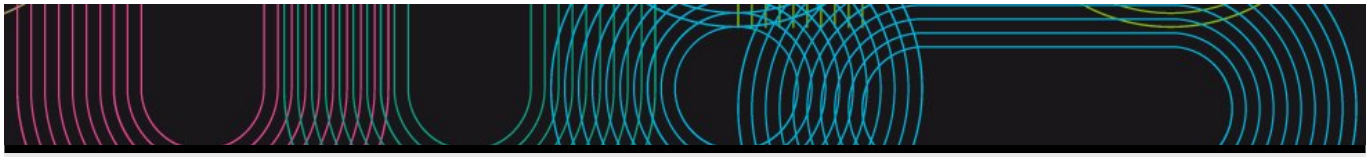

DATOS IDENTIFICATIVOS				
Programación II				
Asignatura	Programación II			
Código	V05G300V01302			
Titulacion	Grado en Ingeniería de Tecnologías de Telecomunicación			
Descriptores	Creditos ECTS	Seleccione	Curso	Cuatrimestre
	6	OB	2	1c
Lengua Impartición	Castellano			
Departamento	Ingeniería telemática			
Coordinador/a	Fernández Masaguer, Francisco			
Profesorado	Blanco Fernández, Yolanda Fernández Masaguer, Francisco Servia Rodríguez, Sandra			
Correo-e	f_masaguer@yahoo.es			
Web	http://www.faitic.es			
Descripción general	El objetivo general de la asignatura es proporcionar al estudiante los fundamentos teóricos y las competencias prácticas que le permitan analizar, diseñar, desarrollar y depurar aplicaciones informáticas siguiendo el paradigma orientado a objetos. Esta es una asignatura eminentemente práctica y en este sentido está orientada al trabajo de los alumnos en la realización de uno o varios proyectos. Para facilitar el desarrollo de los proyectos en la asignatura también se hace una introducción a la Ingeniería del Software. En este sentido no se ocupa de todas las fases generalmente reconocidas en los procesos de desarrollo software que van desde la captura y descripción de requisitos hasta el despliegue de los sistemas, sino que se tratarán principalmente las etapas de análisis, diseño, implementación y depuración. En primer lugar se presentará la ingeniería del software como disciplina imprescindible para el desarrollo de grandes aplicaciones informáticas, mostrando los principales retos a los que se enfrenta y los conceptos básicos que se utilizarán. A continuación se analizarán los elementos del paradigma orientado a objetos utilizando elementos y diagramas UML que serán utilizados por los alumnos en sus desarrollos. Para alcanzar este objetivo general los contenidos que se verán en la asignatura se pueden resumir en los siguientes ítems: El paradigma Orientado a Objetos o Conceptos básicos de la orientación a objetos: clases y objetos o Encapsulación. Principio de ocultación. Conceptos de desacoplamiento y cohesión o Herencia, abstracción, polimorfismo y reutilización o Relaciones entre clases: Generalización, asociación y dependencia o Comunicación entre objetos: métodos, eventos, mensajes o Persistencia. Almacenamiento en ficheros y en bases de datos o Generación, captura y procesamiento de excepciones Introducción a la Ingeniería del Software o Conceptos básicos de la Ingeniería del Software. Reseña histórica o Introducción y concepto de Ciclo de Vida. Estándar ISO/IEC 12207 o Introducción a las metodologías de desarrollo de software. Clasificación o Introducción a los procesos de desarrollo de software orientado a objetos. Métrica v3 y el Proceso Unificado o Fases principales en el desarrollo OO: análisis, diseño, implementación y pruebas o Introducción al lenguaje de modelado UML: estructura e interacción			

Competencias de titulación	
Código	
A6	CG6 Facilidad para el manejo de especificaciones, reglamentos y normas de obligado cumplimiento.
A9	CG9 Capacidad para trabajar en un grupo multidisciplinar y en un entorno multilingüe y de comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.
A59	(CE50/T18) Capacidad de desarrollar, interpretar y depurar programas utilizando los conceptos básicos. de la Programación Orientada a Objetos (POO): clases y objetos, encapsulación, relaciones entre clases y objetos, y herencia.

A60	(CE51/T19) Capacidad de aplicación básica de las fases de análisis, diseño, implementación y depuración de programas en la POO.
A61	(CE52/T20) Capacidad de manejo de herramientas CASE (editores, depuradores).
A62	(CE53/T21) Capacidad de desarrollo de programas atendiendo a los principios básicos de calidad de la ingeniería del software, teniendo en cuenta las principales fuentes existentes en normas, estándares y especificaciones.
B5	CG14 Capacidade para utilizar ferramentas informáticas de procura de recursos bibliográficos ou de información.

Competencias de materia

Resultados previstos en la materia	Resultados de Formación y Aprendizaje	
Comprender los aspectos fundamentales de la Programacion Orientada a Objetos (POO) y llevarlos a la practica usando el lenguaje de programacion mas representativo (Java).	A9 A59	
Introducir en el uso del lenguaje UML, lenguaje estandar de modelado de software, para la realizacion de diagramas de estructura, comportamiento e interacción, fundamental para la documentacion en las fases de análisis y diseño de programas de acuerdo a la POO.	A6 A61 A62	B5
Desarrollar habilidades en el proceso de análisis, diseño, implementacion y depuracion de aplicaciones de acuerdo a la POO teniendo en cuenta los estandares principales y normas de calidad.	A60 A62	
Adquirir madurez en tecnicas de desarrollo y depuracion de programas para permitir el aprendizaje autónomo de nuevas capacidades y lenguajes de programación.	A62	
Adquirir familiaridad con el uso de un entorno moderno de desarrollo de software (Eclipse) para facilitar el diseño, desarrollo y depuración de programas.	A60 A61	

Contenidos

Tema	
1. Introducción al paradigma OO	a. Breve introducción a la asignatura y su organización b. Nacimiento del paradigma c. Bases: clases y objetos d. Conceptos de encapsulación, herencia (generalización), y polimorfismo e. Breve introducción a UML.
2. Encapsulación	a. Clases, interfaces y paquetes b. Métodos y variables miembro. Visibilidad. Resolución de ámbito. c. Método constructor d. Paso de parámetros: punteros y referencias e. Punteros a objetos
3. Herencia	a. Clases derivadas y tipos de herencia b. Clases abstractas c. Herencia múltiple d. Clase object
4. Diseño orientado a objetos	a. Fundamentos de diseño. b. Conceptos básicos de la Ingeniería del Software. c. Utilización de diagramas UML
5. Polimorfismo	a. Sobrecarga y sobreescritura b. Clases abstractas e interfaces c. Clases genéricas
6. Gestión de excepciones	a. Fundamentos de excepciones b. Manipulación de excepciones en Java
7. Recursión	a. Métodos recursivos con devolución de parámetros b. Métodos recursivos sin devolución de parámetros c. Pensando recursivamente

Planificación

	Horas en clase	Horas fuera de clase	Horas totales
Sesión magistral	28	42	70
Resolución de problemas y/o ejercicios	9	9	18
Presentaciones/exposiciones	1	1	2
Resolución de problemas y/o ejercicios de forma autónoma	5	10	15
Proyectos	7	31	38
Pruebas prácticas, de ejecución de tareas reales y/o simuladas.	2	0	2
Estudio de casos/análisis de situaciones	0	1	1
Resolución de problemas y/o ejercicios	2	0	2
Pruebas prácticas, de ejecución de tareas reales y/o simuladas.	2	0	2

*Los datos que aparecen en la tabla de planificación son de carácter orientativo, considerando la heterogeneidad de alumnado

Metodologías	
	Descripción
Sesión magistral	Clases que combinarán la exposición de los conceptos a tratar en la asignatura con la realización de pequeños ejercicios. Éstos podrán ser resueltos por el docente o por los propios alumnos individualmente y/o en grupo. El objetivo es fomentar el debate en la clase y reforzar la adquisición de destrezas.
Resolución de problemas y/o ejercicios	En el laboratorio, el profesor planteará pequeños retos que serán resueltos colectivamente para que se puedan debatir los conceptos subyacentes, las diferentes opciones de resolución y que los alumnos adquieran las destrezas objetivo de la asignatura.
Presentaciones/exposiciones	Los alumnos expondrán a sus compañeros en el laboratorio el diseño planteado para solucionar el sistema software objetivo del proyecto que han de llevar a cabo durante la segunda parte del curso. Comparando las diferentes propuestas se plantearán las mejores opciones y servirá como realimentación para, si es oportuno, mejorar los diseños realizados.
Resolución de problemas y/o ejercicios de forma autónoma	Los alumnos resolverán de forma autónoma los problemas que el profesor le plantee en el laboratorio. Las soluciones y las dudas que surjan al abordar dichos problemas serán puestas en común para consensuar la mejor forma de resolución.
Proyectos	Los alumnos implementarán el sistema software planteado por el profesor. Dispondrá para ello de la segunda parte del curso combinando trabajo presencial en el laboratorio con el trabajo fuera del laboratorio.

Atención personalizada	
Metodologías	**Descripción**
Resolución de problemas y/o ejercicios	La atención individualizada se articulará con el seguimiento del trabajo de cada alumno, monitorizando las soluciones que propone para cada problema planteado en las prácticas de laboratorio, la exposición de las mismas que realice a sus compañeros y el seguimiento del proyecto software que debe implementar.
Presentaciones/exposiciones	La atención individualizada se articulará con el seguimiento del trabajo de cada alumno, monitorizando las soluciones que propone para cada problema planteado en las prácticas de laboratorio, la exposición de las mismas que realice a sus compañeros y el seguimiento del proyecto software que debe implementar.
Proyectos	La atención individualizada se articulará con el seguimiento del trabajo de cada alumno, monitorizando las soluciones que propone para cada problema planteado en las prácticas de laboratorio, la exposición de las mismas que realice a sus compañeros y el seguimiento del proyecto software que debe implementar.
Resolución de problemas y/o ejercicios de forma autónoma	La atención individualizada se articulará con el seguimiento del trabajo de cada alumno, monitorizando las soluciones que propone para cada problema planteado en las prácticas de laboratorio, la exposición de las mismas que realice a sus compañeros y el seguimiento del proyecto software que debe implementar.

Evaluación		
	Descripción	Calificación
Proyectos	Los alumnos, organizados en grupos de 2 personas, entregarán el proyecto software propuesto durante la primera semana de Diciembre (del 2 al 6 de Diciembre). Éste constará de su diseño final (diagramas UML), el código y la documentación generada explicativa de la implementación. Que el código entregado pueda ser compilado y ejecutado en los equipos de los laboratorios docentes es llave para superar esta evaluación. Los docentes valorarán en igual proporción el funcionamiento del código entregado y el diseño utilizado para la implementación. Con esta prueba se evaluaran las competencia CE53, CE50	15
Pruebas prácticas, de ejecución de tareas reales y/o simuladas.	Durante la segunda semana del mes de Diciembre (del 9 al 13 de Diciembre), los alumnos tendrán una entrevista con el profesor en el horario de laboratorio en el que deberán responder diferentes cuestiones en relación al proyecto software entregado (e.g. justificar decisiones de diseño y proponer soluciones para abordar determinadas modificaciones en el proyecto planteado). Los dos miembros de cada grupo deben estar obligatoriamente presentes en dicha entrevista. Las cuestiones planteadas en la misma deberán ser respondidas individualmente para poder constatar la autoría, el grado de entendimiento e implicación del alumno en el proyecto desarrollado. En caso de que el alumno no acredite los aspectos anteriores, el profesor podrá exigir la realización de un examen de programación individual en el laboratorio docente en la fecha oficial aprobada por la Junta de Escuela a tal fin.	15

Estudio de casos/análisis de situaciones	Los alumnos, organizados en grupos de 2 personas, habrán de entregar el diseño de un proyecto software. Se entregará en la primera semana del mes de Noviembre (del 4 al 7 de Noviembre). Con esta prueba se evaluaran las competencia CE51, CE52	10
Resolución de problemas y/o ejercicios	Examen escrito e individual, realizado en la fecha aprobada por Junta de Escuela para ello, que constará de la combinación de los siguientes tipos de preguntas: resolución de problemas, cuestiones breves para resolver aplicando los conceptos teóricos explicados en clase, justificar razonadamente si una o varias afirmaciones son verdaderas o falsas, pequeños tests sobre aspectos teóricos y de aplicación. No se permite la utilización de apuntes, libros ni colecciones de problemas. El número y la combinación de dichas preguntas se fijará para cada examen en particular. Con esta prueba se evaluaran las competencia CE50, CE51,CE53	50
Pruebas prácticas, de ejecución de tareas reales y/o simuladas.	En la semana del 21 al 25 de Octubre, los alumnos, organizados en grupos de 2 personas, entregarán las prácticas de iniciación en Java propuestas en el laboratorio.	10

Otros comentarios sobre la Evaluación

Existen dos modalidades en la evaluación de la asignatura: evaluación continua (EC) y evaluación tradicional (ET). En cualquiera de los dos esquemas, el alumno superará la asignatura si consigue al menos 5 puntos (sobre un total de 10).

Los alumnos deberán elegir una de las dos modalidades teniendo en cuenta las siguientes restricciones:

- La EC incluye las 5 pruebas descritas anteriormente.
- Tanto por EC como por ET, los alumnos deberan realizar un proyecto de laboratorio. Para facilitar la elección de EC o ET los alumnos dispondrán en Faitic del proyecto a realizar a partir del día 20 de Septiembre.
- En ET el proyecto se realizará de forma individual.
- Los alumnos que opten por la EC deberán entregar en la semana del 4 al 7 de Noviembre, el diseño UML del proyecto planteado en el laboratorio (correspondiente a la 3ª prueba de evaluación). Mediante dicha entrega los alumnos se comprometen a seguir la EC y renuncian a la ET. Desde ese momento, estos estudiantes no podrán figurar como "No presentados".
- Los alumnos que no entreguen el diseño UML del proyecto en la semana del 4 al 7 de Noviembre, renuncian a la EC, de modo que serán evaluados mediante el mecanismo de ET. No existe la posibilidad de sumarse a la EC en las siguientes pruebas intermedias.
- Las pruebas de EC no serán en ningún caso recuperables, no pudiendo repetirse fuera de las fechas estipuladas por los docentes.
- No se guardarán calificaciones (de pruebas de EC ni de proyectos prácticos o exámenes finales) de un curso a otro.
- La EC sólo se aplicará en la convocatoria de Enero, en el resto de convocatorias rige únicamente la ET.

Los alumnos que opten por la EC serán evaluados con arreglo a las pruebas descritas anteriormente:

- Prácticas de iniciación en Java (10%). En grupos de 2 alumnos. Se corresponde con la prueba 5 descrita en el apartado "Evaluación".
- Proyecto (40%). En grupos de 2 alumnos. Se desglosa en tres partes: diseño(10%), implementación (15%) y entrevista personal (15%). Estas partes se corresponden con las pruebas 3, 1 y 2, respectivamente, descritas en la sección previa.
- Examen escrito (50%). Se corresponde con la prueba 4 descrita anteriormente.

Los alumnos que opten por la ET serán evaluados como sigue:

- Un examen escrito (cuya descripción coincide con la prueba 4 de la evaluación continua). El resultado de este examen supondrá un 50% de la calificación final.
- La realización de un proyecto software que constará de diseño (diagramas UML), el código y la documentación generada explicativa de la implementación. Que el código entregado pueda ser compilado y ejecutado en los equipos de los laboratorios docentes es llave para superar esta evaluación. Los docentes valorarán a partes iguales el funcionamiento del código entregado y el diseño utilizado para la implementación. La evaluación de esta prueba supondrá un 30% de la calificación final. Este proyecto deberá ser entregado individualmente la semana del 2 al 5 de Diciembre del periodo docente.
- La realización de una entrevista en la que el alumno deberá responder a las cuestiones planteadas por los docentes en relación a las decisiones de diseño tomadas, así como a la manera de abordar soluciones para posibles modificaciones del proyecto planteado. Dicha entrevista tendrá lugar en el laboratorio la semana del 9 al 13 de

Diciembre del periodo docente y supondrá un 20% de la calificación final.

Para la **convocatoria de Julio** no rige la EC, por lo que todos los alumnos se acogerán a la modalidad de ET que será como sigue:

- Un examen escrito (cuya descripción coincide con la prueba 4 de la EC). El resultado de este examen supondrá el 50% de la calificación final. No se permitirá material de apoyo.
- La realización de un examen de programación individual en el laboratorio que tendrá lugar en la fecha fijada por la Junta de Escuela para ello. La evaluación de esta prueba supondrá un 50% de la calificación final.

Fuentes de información

[1] *Absolute Java*. Walter Savitch, 4ª edición. 2010, Pearson.

[3] *Java: How to program*/*Java: cómo programa*r. P. Deitel y H. Deitel, 9ª edición. 2011, Pearson.

[2] *The Java Tutorial. A Short course on the basics*. Sharon Zakhour, Scott Hommel, Jacob Royal, Isaac Rabinovitch, Tom Risser, Mark Hoeber, 4ª edición. 2006, Prentice-Hall.

[13] *Ingeniería del Software orientada a objetos con UML, Java e Internet*. Alfredo Weitzenfeld. 2005, Thomson.

[15]Object-Oriented Analysis and Design with Applications. Grady Booch. 2011, Addison Wesley.
[16] UML Distilled: A Brief Guide to the Standard Object Modeling Language. Martin Fowler. 3ª edición.

Recomendaciones

Asignaturas que se recomienda haber cursado previamente

Programación I/V05G300V01205