

1.- Datos de la Asignatura

Titulación	INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN					
Centro	ESCUELA POLITÉCNICA SUPERIOR DE ZAMORA					
Denominación	PROGRAMACIÓN ORIENTADA A OBJETOS			Código	16911	
Plan	2003		Ciclo		Curso	3º
Carácter ¹	B		Periodicidad ²	C1		
Créditos LRU	T	3	P	3	Créditos ECTS	
Área	LENGUAJES Y SISTEMAS INFORMÁTICOS					
Departamento	INFORMÁTICA Y AUTOMÁTICA					
Aula / Horario / grupo	P-117		MIÉRCOLES: 17:00:18:00 JUEVES: 17:00:18:00	ÚNICO		
Laboratorio/ Horario / grupo						
Informática / Horario / grupo	Aula INFORMÁTICA V Aula INFORMÁTICA V		LUNES: 12:30-14:30 MARTES: 10:30-12:30	I II		
Plataforma Virtual	Plataforma: Eudored-Moodle					
	URL de Acceso: http://eudored.usal.es					

¹ Troncal, Obligatoria, Optativa (abreviatura T, B, O)

² Anual, 1º Cuatrimestre, 2º Cuatrimestre (A, C1, C2).

Datos del profesorado*

Profesor Responsable / Coordinador	JUAN CARLOS MATOS FRANCO JESÚS ÁNGEL ROMÁN GALLEGO				
Departamento	INFORMÁTICA Y AUTOMÁTICA				
Área	LENGUAJES Y SISTEMAS INFORMÁTICOS				
Centro	ESCUELA POLITÉCNICA SUPERIOR DE ZAMORA				
Despacho	218-Edificio Administrativo		Grupo / s		
Horario de tutorías	Martes 12:00 a 15:00; Jueves 12:00 a 15:00				
URL Web					
E-mail	jcmatos@usal.es zjarg@usal.es		Teléfono	980545000 ext. 3636	

* Caso de que sea una asignatura impartida por más de un docente.

*Esta tabla se repetirá tantas veces como sea necesario, en el caso de que sean varios docentes los responsables de impartir la materia, dedicando una tabla para cada docente.

2.- Sentido de la materia en el plan de estudios*

Bloque formativo al que pertenece la materia

Papel de la asignatura dentro del Bloque formativo y del Plan de Estudios.

Iniciar al alumno en las bases de la Programación Orientada a Objetos (POO), explicándole los conceptos básicos (abstracción, encapsulación y ocultación de la información; clases, subclasses, herencia y polimorfismo) y enseñándole a diseñar problemas relativos a objetos, no intentando “traducirlos” a partir de otros esquemas ya realizados con otros paradigmas de programación.

Perfil profesional.

El principal interés de la materia es mostrar a los alumnos la posibilidad de que ellos mismos creen sus propias aplicaciones desde el paradigma de la orientación a objetos, muy utilizado en la actualidad. Para ello se emplea un lenguaje de programación de fácil aprendizaje como es Java, que les resulta muy sencillo pues su sintaxis es similar a la del lenguaje C, que es el que han venido empleando en las asignaturas de programación en cursos anteriores. Además se realiza una visión general de la aplicación de los conceptos orientados a objetos en otros lenguajes de programación de uso habitual, como son C++, PHP o C#. En la vida profesional estos conocimientos permiten el abordaje de problemas de una manera directa y estructurada, empleando un diseño adecuado.

**Esta información se puede obtener, en la mayoría de los casos, en los libros blancos de la ANECA para cada titulación. http://www.aneca.es/modal_eval/conver_docs_titulos.html.*

3.- Recomendaciones previas*

Se recomienda haber cursado previamente las siguientes asignaturas:

- Programación I.
- Estructuras de datos y algoritmos.

También es conveniente haber realizado antes la asignatura de Diseño de Bases de Datos.

** Requisitos previos o mínimos que en algunas materias son necesarios para cursar la asignatura (asignaturas previas, conocimientos concretos, habilidades y destrezas determinadas,...)*

Datos Metodológicos

4.- Objetivos de la asignatura (Generales y Específicos)

Objetivos:

- Adquirir conocimientos genéricos sobre lenguajes de programación orientados a objetos.
- Aplicar estos conocimientos al aprendizaje de un lenguaje de programación específico como es Java.
- Realizar la descomposición de los problemas con un diseño orientado a objetos y codificarlo en el lenguaje Java.
- Una vez adquiridos unos conocimientos básicos, ser capaces de abordar el desarrollo de una aplicación práctica.

5.- Contenidos

Teoría

- I. Introducción a la asignatura
 - 1- Introducción
- II. Aspectos básicos
 - 2- Clases y Objetos
 - 3- Relaciones entre clases
 - 4- Herencia y polimorfismo
- III. Aspectos avanzados
 - 5- Principios del diseño orientado a objetos
 - 6- Patrones de diseño

Práctica

Se proponen 8 prácticas relacionadas con los temas tratados en la asignatura, aplicando Java y UML para desarrollar los conceptos teóricos de la misma:

- PRÁCTICA 1: Máquinas virtuales. Introducción a la POO y a Java.
- PRÁCTICA 2: Diseño con UML. Relaciones entre objetos.
- PRÁCTICA 3: Clases y objetos en Java.
- PRÁCTICA 4: Herencia y jerarquías de clases.
- PRÁCTICA 5: Caso de Estudio: Proyecto de diseño básico.
- PRÁCTICA 6: Interfaces.
- PRÁCTICA 7: Patrones de diseño.
- PRÁCTICA 8: Caso de Estudio: Proyecto de diseño avanzado.

6.- Competencias a adquirir*

Competencias Específicas. (En relación a los conocimientos, habilidades. y actitudes: conocimientos destrezas, actitudes...)

Con esta asignatura se pretende introducir al alumno en el desarrollo de pequeñas aplicaciones con el paradigma de orientación a objetos.

COMPETENCIAS ACADÉMICAS:

- Conocer qué es un lenguaje de programación orientado a objetos y los distintos tipos que existen.
- Diseñar programas sencillos para la resolución de problemas.

CONOCIMIENTOS DISCIPLINARES (hacer):

- Manejar un entorno profesional de desarrollo con Java.
- Implementar los ejemplos propuestos empleando la sintaxis aprendida.

COMPETENCIAS PROFESIONALES (saber hacer);

- Ante un problema determinado, saber abordarlo de manera estructurada con el enfoque de orientación a objetos, resolviéndolo en papel e implementándolo posteriormente en un lenguaje de programación como es Java.
- Desarrollar una aplicación a partir de una especificación de requisitos.

Transversales: (Competencias Instrumentales: <cognitivas, metodológicas, tecnológicas o lingüísticas>; Competencias Interpersonales <individuales y sociales>; o Competencias Sistémicas. <organización, capacidad emprendedora y liderazgo>

Competencias Instrumentales.

- Capacidad de análisis y síntesis. A la hora de desarrollar cualquier proyecto software, sea cual se su escala, se necesita cierta capacidad de análisis y síntesis, de manera que a partir de unos requisitos que especifiquen qué es lo que se desea realizar se pueda desarrollar desde el punto de vista de la programación. Con esto se pretende que los alumnos sean capaces de, a partir de un enunciado especificado, analizarlo, resolverlo y codificarlo como un programa de computador.
- Capacidad de organización y planificación. Una de las partes importantes a la hora de desarrollar cualquier tipo de proyecto es la de ser capaz de planificar y organizar las distintas tareas. Mediante la asignación de trabajos a los alumnos para desarrollar en grupo se ven obligados a realizar una planificación del trabajo a realizar.
- Comunicación oral y escrita en la lengua nativa. Además de realizar un programa para computador deben de ser capaces de explicar adecuadamente qué es lo que hace.
- Resolución de problemas. El alumno debe ser capaz de resolver una serie de problemas de manera programática.

Competencias interpersonales.

- Trabajo en equipo. El alumno debe aprender a realizar los trabajos asignados en grupo, organizando, colaborando y distribuyendo las tareas adecuadamente.
- Habilidades en relaciones interpersonales. El trabajo en grupo debe fomentar las relaciones interpersonales, tanto con los miembros de su grupo de trabajo como del resto de grupos.
- Razonamiento crítico. El alumno debe ser capaz tanto de evaluar su propio desempeño en la materia como el de sus compañeros.
- Compromiso ético. El trabajo en equipo requiere un compromiso ético con el resto de compañeros que forman el grupo.

Competencias sistémicas

- Aprendizaje autónomo. Mediante el uso del material didáctico dispuesto el alumno debe conseguir un aprendizaje autónomo de manera que él mismo lleve el ritmo.

- Creatividad. Mediante la realización de un trabajo/proyecto de desarrollo se fomenta la creatividad, dado que los enunciados son meramente informativos de lo que se debe de realizar dejando libre al alumno para que cree y desarrolle a su gusto, siempre dentro de unas pautas.
- Iniciativa y espíritu emprendedor. Mediante el trabajo se fomenta que no solamente se queden en la consecución de los requisitos mínimos, sino que sean capaces de ir mas allá, incluso de cambiar estos requisitos.
- Motivación por la calidad. El alumno debe darse cuenta que siguiendo las pautas y recomendaciones, y realizando las cosas con cuidado, el resultado final será mejor.
- Sensibilidad hacia temas medio ambientales. Mediante el uso de soportes informáticos y medios de intercambio de información digitales el alumno debe de ver que se produce un ahorro de papel y materias primas.

*Según la clasificación establecida por la ANECA, esta tabla puede ser más adecuada para las asignaturas que ya están adaptadas al modelo del EEES. En los documentos recogidos por la ANECA para cada titulación, se especifican las competencias tanto específicas como transversales o genéricas. Esta relación de competencias se puede consultar en: http://www.aneca.es/modal_eval/conver_docs_titulos.html

7.- Metodologías

Clases Teóricas:

Hay dos días a la semana. En el primero de ellos se explican los conceptos teóricos y el segundo se desarrollan ejemplos y/o ejercicios que resuelven y debaten los alumnos en grupo.

Clases Prácticas:

Principalmente se realizan prácticas de forma tutorizada (se hacen los ejercicios con los alumnos de forma guiada). En la parte final de la clase se proponen ejercicios que resuelven ellos solos. Algunos días las prácticas se realizan como desarrollo de casos de estudio que diseñan y resuelven en grupo y que se entregan (opcionalmente) para su evaluación.

Interacción con el alumno:

Se fomentará la interacción del alumno por diferentes vías:

- *Clases presencial:* dado el carácter práctico de la asignatura el profesor estará presente en el aula para resolver las posibles dudas de los alumnos y guiarlos en la realización de los ejercicios y el trabajo final.
- *Tutorías:* los alumnos podrán acudir a tutorías para cualquier consulta relativa a la materia.
- *Espacio virtual:* se dispondrá de la herramienta Eudored/Moodle para el intercambio de información con los alumnos (apuntes, ejercicios, etc.) y como medio de comunicación (foros y chat). Las entregas de trabajos también se realizarán bajo esta plataforma.

8.- Previsión de Técnicas (Estrategias) Docentes*

Opcional para asignaturas de cualquier curso			
	Horas presenciales	Horas no presenciales	Horas totales
Clases magistrales			
Clases prácticas			
Seminarios			

Exposiciones y debates			
Tutorías			
Actividades no presenciales			
Preparación de trabajos			
Otras actividades			
Exámenes			
TOTAL			

Esta tabla está pensada para aquellas asignaturas que **no han sido planificadas teniendo en cuenta los créditos ECTS.*

8.- Previsión de Técnicas (Estrategias) Docentes*

Opcional para asignaturas de 1er curso				
	Horas presenciales.	Horas no presenciales.	Horas de trabajo autónomo del alumno	Horas totales
Clases magistrales	10		20	30
Clases prácticas	30		45	75
Seminarios				
Exposiciones y debates	8		8	16
Tutorías			4	4
Actividades no presenciales		4		4
Preparación de trabajos		4	20	24
Otras actividades				
Exámenes	3			3
TOTAL	51	8	97	156

**Para las asignaturas cuya estructura y organización se haya realizado en base a los créditos ECTS.*

9.- Recursos

Libros de consulta para el alumno
<p>TEORÍA</p> <ul style="list-style-type: none"> • BOOCH, G., RUMBAUGH, J. Y JACOBSON, I. El lenguaje unificado de modelado. Addison Wesley Iberoamericana. 1999. • GAMMA, E., HELM, R., JOHNSON, R. Y VLISSIDES, J. Patrones de Diseño, Ed. Addison Wesley. 2002. • MUÑOZ CARO, C., NIÑO RAMOS, A. Y VIZCAÍNO BARCELÓ, A. Introducción a la programación con orientación a objetos. Prentice Hall. 2002. • RUMBAUGH, J., BLAHA, M., PREMERLANI, W., EDDY, F. and LORENSEN, W. Modelado y diseño orientados a objetos. Prentice Hall, 1996. <p>PRÁCTICA</p> <ul style="list-style-type: none"> • BARNES, D. Y KÖLLING, M. Objects First with Java - A Practical Introduction using BlueJ, 2º Ed. Prentice Hall / Pearson Education. 2004. • CEBALLOS, F.J. Java 2. Curso de programación, 3ª edición. RA-MA Editorial. 2007.

- ECKEL, B. Thinking in Java, 3º Ed. Prentice Hall. 2003.
- STELTING, S. Y MAASSEN, O. Patrones de diseño aplicados a Java, Ed. Prentice Hall. 2003.

Otras referencias bibliográficas, electrónicas o cualquier otro tipo de recurso.

Alice
<http://www.alice.org/>
 ArgoUML
<http://argouml.tigris.org/>
 BlueJ
<http://www.bluej.org/download/download.html>
 Documentación de Java
<http://java.sun.com/javase/6/docs/>
 Eclipse
<http://www.eclipse.org/downloads/>
 GreenFoot
<http://www.greenfoot.org/>
 Java
<http://java.sun.com/javase/downloads/>
 Jeliot
<http://cs.joensuu.fi/jeliot/>
 NetBeans
<http://www.netbeans.org/>
 Robocode
<http://robocode.sourceforge.net/>

10.- Evaluación

Consideraciones Generales

La evaluación se realizará teniendo en cuenta:

- Evolución en clase del alumno.
- Trabajos realizados y defensa de los mismos.
- Examen teórico-práctico.

Criterios de evaluación

Suspenso (0-4,9):

- El alumno no ha adquirido los conocimientos mínimos, ni la pericia mínima para aprobar la signatura.

Aprobado (5-6,9):

- El alumno comprende los fundamentos de la programación orientada a objetos.
- Es capaz de realizar la mayoría de los ejercicios propuestos.

Notable (7,0-8,9):

- El alumno comprende los fundamentos de la programación orientada a objetos.
- Es capaz de realizar los ejercicios propuestos.
- Ha mostrado interés en los temas propuestos.
- Ha sido capaz de incorporar ideas nuevas con mayor o menor acierto.

Sobresaliente (9,0-10):

- El alumno comprende los fundamentos de la programación orientada a objetos.
- Es capaz de realizar los ejercicios propuestos.
- Ha mostrado interés en los temas propuestos.
- Ha sido capaz de incorporar gran cantidad de ideas nuevas con acierto, así como de proponer modificaciones, posibles ampliaciones, etc.

Instrumentos de evaluación

Los instrumentos para la evaluación son:

- Evolución en clase del alumno.
- Trabajos realizados y defensa de los mismos.
- Examen teórico-práctico.

Recomendaciones para la evaluación.

La asistencia a clase y la participación del alumno unido al trabajo continuo, permiten superar sin dificultad la asignatura.

Recomendaciones para la recuperación.