

GUÍA DOCENTE
EXPERIENCIA PILOTO DE APLICACIÓN DEL SISTEMA EUROPEO DE CRÉDITOS (ECTS)
UNIVERSIDADES ANDALUZAS

TITULACIÓN: INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN
(MODALIDAD SEMIVIRTUAL)

DATOS BÁSICOS DE LA ASIGNATURA

NOMBRE: ESTRUCTURAS DE DATOS Y DE LA INFORMACIÓN

CÓDIGO: 904

AÑO DE PLAN DE ESTUDIO: 2004

TIPO (troncal/obligatoria/optativa): Troncal

Créditos totales (LRU/ECTS):
12

Créditos LRU/ECTS teóricos: 6

Créditos LRU/ECTS prácticos: 6

CURSO: 1º

CUATRIMESTRE(S): Anual

CICLO: 1º

EQUIPO DOCENTE

Responsable / Coordinador de la asignatura:

NOMBRE: Raúl Giráldez Rojo

CENTRO/DEPARTAMENTO: Escuela Politécnica Superior / Deporte e Informática

ÁREA: Lenguajes Y Sistemas Informáticos

CATEGORÍA: Profesor Titular de Universidad

POD		GRUPOS	CRÉDITOS	TOTAL
	Gran Grupo		2,4	
	Grupo de Docencia		2	
	Activ. Dirigidas		0,8	

HORARIO DE TUTORÍAS:

Nº DESPACHO: 11.2.10

E-MAIL: rgirroj@upo.es

TF: +34 954977998

URL WEB: www.upo.es/eps/giraldez

LA ASIGNATURA EN EL PROGRAMA FORMATIVO

1. DESCRIPTOR.

Comprender los fundamentos técnicos de la programación.

Conocer y utilizar los lenguajes de programación orientada a objetos para el desarrollo de aplicaciones informáticas y sistemas software.

Conocer las estructuras de datos complejas, sus aplicaciones y propiedades.

2. UBICACIÓN EN EL PROGRAMA FORMATIVO.

2.1. PRERREQUISITOS:

Ninguno.

2.2. CONTEXTO DENTRO DE LA TITULACIÓN:

La asignatura de EDI introduce a los alumnos en las técnicas de programación orientada a objetos desde un punto de vista técnico, haciendo especial énfasis en el diseño y utilización de las estructuras de datos y tipos abstractos de datos. Esta asignatura, junto con MTP1 y MPT2, conforman una completa formación en la creación de aplicaciones informáticas y dotará al alumno de conocimientos de programación necesarios tanto en asignaturas de cursos superiores como en el mercado laboral.

2.3. RECOMENDACIONES:

Ninguna.

3. LA ASIGNATURA EN LA ADQUISICIÓN DE COMPETENCIAS.

3.1. COMPETENCIAS TRANSVERSALES/GENÉRICAS:

- Conocimientos generales básicos.
- Solidez en los conocimientos básicos de la profesión.
- Habilidades elementales en informática.
- Resolución de problemas.
- Capacidad de aprender.
- Habilidad para trabajar de forma autónoma.

3.2. COMPETENCIAS ESPECÍFICAS:

- **Cognitivas (Saber):** Enseñar al alumno unos conocimientos generales básicos sobre programación orientada a objetos: clases y objetos, interfaces, reutilización y adaptación de código. Enseñarle la utilización de estructuras de datos y tipos abstractos de datos complejos.
- **Procedimentales/Instrumentales (Saber hacer):** Enseñar al alumno las técnicas básicas para la resolución de problemas mediante programas (algoritmos y estructuras de datos). Mentalizar al alumno de que la construcción de programas sigue una metodología con procesos sistemáticos para alcanzar el objetivo.
- **Actitudinales (Ser):** Fomentar la disciplina en la programación, siguiendo planteamientos que desarrollen la capacidad analítica del alumno para enfrentarse a problemas reales.

4. OBJETIVOS.

- Estudiar el paradigma de la programación orientada a objetos mediante un lenguaje de programación concreto.
- Conocer estructuras de datos complejas y esquemas algorítmicos básicos de utilización de las mismas.
- Desarrollar una metodología de programación.

DISTRIBUCIÓN DEL TRABAJO PRESENCIAL.			
	Gran Grupo	Grupo de Docencia	Actividades dirigidas (seminarios)
Nº de grupos	1	2	2
Nº de horas	24	20	8
Nº de sesiones	16	10	4

5. METODOLOGÍA.

NÚMERO TOTAL DE HORAS DE TRABAJO DEL ALUMNO: 340

PRIMER SEMESTRE: 175 horas de trabajo

Nº de Horas:

- Enseñanzas básicas (Gran Grupo): 13,5
- Enseñanzas prácticas y de desarrollo (Grupo de Docencia): 10
- Actividades académicas dirigidas (Seminarios-Grupo de Trabajo): 4
- Tutorías especializadas (presenciales o virtuales):
 - A) Colectivas:
 - B) Individuales: 6
- Trabajo personal autónomo:
 - A) Horas de estudio de enseñanzas básicas: 33,5
 - B) Horas de estudio-preparación de las enseñanzas prácticas y de desarrollo: 80
 - C) Horas de trabajo personal o en grupo derivadas de las actividades académicas dirigidas: 24
- Otras actividades (visitas, excursiones, etc.)
- Realización de pruebas de evaluación y/o exámenes:
 - A) Prueba de evaluación y/o exámenes escritos: 4
 - B) Pruebas de evaluación y/o exámenes orales (control del Trabajo Personal)

SEGUNDO SEMESTRE: 165 horas de trabajo

Nº de Horas:

- Enseñanzas básicas (Gran Grupo): 10,5
- Enseñanzas prácticas y de desarrollo (Grupo de Docencia): 10
- Actividades académicas dirigidas (Seminarios-Grupo de Trabajo): 4
- Tutorías especializadas (presenciales o virtuales):
 - A) Colectivas:
 - B) Individuales: 6
- Trabajo personal autónomo:
 - A) Horas de estudio de enseñanzas básicas: 26,5
 - B) Horas de estudio-preparación de las enseñanzas básicas y de desarrollo: 80
 - C) Horas de trabajo personal o en grupo derivadas de las actividades académicas dirigidas: 24
- Otras actividades (visitas, excursiones, etc.)
- Realización de pruebas de evaluación y/o exámenes:
 - A) Pruebas de evaluación y/o exámenes escritos: 4
 - B) Pruebas de evaluación y/o exámenes orales (control del Trabajo Personal):

6. TÉCNICAS DOCENTES. (Señale con una X las técnicas que va a utilizar en el desarrollo de su asignatura. Puede señalar más de una).

Sesiones académicas teóricas: X	Exposición y debate: X	Tutorías especializadas: X
Sesiones académicas prácticas: X	Visitas y excursiones:	Controles de lecturas obligatorias:

Otras (especificar):

DESARROLLO Y JUSTIFICACIÓN:

Todas las Actividades Prácticas y de Desarrollo así como las Actividades Académicas Dirigidas llevarán asociada una documentación que se proporcionará al alumno a través de la página Web de la asignatura dentro del aula virtual WebCT.

7. BLOQUES TEMÁTICOS. (Dividir el temario en grandes bloques temáticos; no hay número mínimo ni máximo).

- BLOQUE 1: PROGRAMACIÓN ORIENTADA A OBJETOS
- BLOQUE 2: ESTRUCTURAS DE DATOS

8. BIBLIOGRAFÍA.

8.1 GENERAL:

1. Java 2 Manual de Programación. Luis Joyanes Aguilar, Matilde Fernández Azuela. McGraw-Hill 2001.
2. Programación en Java 2. Jesús Sánchez Allende y otros. McGraw-Hill, 2005.
3. Programación, Algoritmos y Ejercicios Resueltos en Java. David Camacho (coordinador). Prentice Hall Pearson, 2003.
4. Thinking in Java, 3rd Edition. Bruce Eckel. Prentice Hall, 2002. <http://www.mindview.net/Books/TIJ/>
5. Documentación Java en Sun Microsystems: <http://java.sun.com/j2se/1.5.0/docs/>

8.2 ESPECÍFICA: (con remisiones concretas en lo posible)

Tema 1: Introducción a la Orientación a Objetos en Java

- Java 2 Manual de Programación. Luis Joyanes Aguilar, Matilde Fernández Azuela. McGraw-Hill 2001. Capítulos 3, 6 y 7.
- Programación, Algoritmos y Ejercicios Resueltos en Java. David Camacho (coordinador). Prentice Hall Pearson, 2003. Capítulo 1.

Tema 2: Interfaces

- Programación en Java 2. Algoritmos, Estructuras de Datos y Programación Orientada a Objetos. Luis Joyanes Aguilar, Ignacio Zahonero Martínez. Páginas 337 a 342.
- Java in a Nutshell, 4ª Ed. David Flanagan. Páginas 117 a 122.
- Java 2. Manual de Programación. Luis Joyanes Aguilar, Matilde Fernández Azuela. Páginas 135 a 139.
- Learning Java. 2ª Ed. Patrick Niemeyer, Johnathan Kundsén. Páginas 151 a 156.

Tema 3: Clases: Implementación de Interfaces

- Programación en Java 2. Jesús Sánchez Allende y otros. McGraw-Hill, 2005. Páginas 35 a 68.
- Programación en Java 2. Luis Joyanes Aguilar, Ignacio Zahonero Martínez. McGraw-Hill, 2002. Páginas 274 a 300.

Tema 4: Reutilización y adaptación: Herencia vs. Composición

- Programación en Java 2. Luis Joyanes Aguilar, Ignacio Zahonero Martínez. McGraw-Hill, 2002. Páginas 308 a 325.
- Programación, Algoritmos y Ejercicios Resueltos en Java. David Camacho (coordinador). Prentice Hall Pearson, 2003. Páginas 279 a 286.
- Programación en Java 2. Jesús Sánchez Allende y otros. McGraw-Hill, 2005. Páginas 148 y 150 a 151.

Tema 5: Clases de Fundamentales de Java (java.lang)

- **String:** Programación en Java 2. Luis Joyanes Aguilar, Ignacio Zahonero Martínez. McGraw-Hill, 2002. Páginas 388 a 417.
- **Excepciones:** Programación en Java 2. Luis Joyanes Aguilar, Ignacio Zahonero Martínez. McGraw-Hill, 2002. Páginas 632 a 658.
- **Envoltura:** Programación en Java 2. Luis Joyanes Aguilar, Ignacio Zahonero Martínez. McGraw-Hill, 2002. Páginas 141 a 145.

Tema 6: Tablas

- Programación en Java 2. Luis Joyanes Aguilar, Ignacio Zahonero Martínez. McGraw-Hill, 2002. Páginas 347 a 381.
- Programación en Java 2. Jesús Sánchez Allende y otros. McGraw-Hill, 2005. Páginas 185 a 187.
- Documentación de la API de Java: <http://java.sun.com/j2se/1.5.0/docs/api/java/util/Arrays.html>

Tema 7: Introducción al Java Collection Framework

- Página principal de JCF: <http://java.sun.com/j2se/1.5.0/docs/guide/collections/index.html>
- Tutorial de JCF: <http://java.sun.com/docs/books/tutorial/collections/intro/index.html>
- Thinking in Java, 3rd Edition. Bruce Eckel. Prentice Hall, 2002. Capítulo 11.
<http://www.mindview.net/Books/TIJ/>

Tema 8: Listas mediante la interfaz List

- Tutorial sobre List: <http://java.sun.com/docs/books/tutorial/collections/interfaces/list.html>
- Thinking in Java, 3rd Edition. Bruce Eckel. Prentice Hall, 2002. Capítulo 11.
<http://www.mindview.net/Books/TIJ/>

Tema 9: Conjuntos mediante las interfaces Set y SortedSet

- Tutoriales sobre Set y SortedSet: <http://java.sun.com/docs/books/tutorial/collections/interfaces/set.html>
<http://java.sun.com/docs/books/tutorial/collections/interfaces/sorted-set.html>
- Especificaciones: <http://java.sun.com/j2se/1.5.0/docs/api/java/util/Set.html>
<http://java.sun.com/j2se/1.5.0/docs/api/java/util/HashSet.html>
<http://java.sun.com/j2se/1.5.0/docs/api/java/util/SortedSet.html>
<http://java.sun.com/j2se/1.5.0/docs/api/java/util/TreeSet.html>
- Thinking in Java, 3rd Edition. Bruce Eckel. Prentice Hall, 2002. Capítulo 11.
<http://www.mindview.net/Books/TIJ/>

9. TÉCNICAS DE EVALUACIÓN.

- La evaluación se basará principalmente en los conocimientos adquiridos tanto en clase de teoría como en aula de informática. La participación también será evaluada.
- Teniendo en cuenta los criterios de evaluación y calificación expuesta abajo, la evaluación de cada parte de la asignatura se realizará según las siguientes indicaciones:
 - Las Enseñanzas Básicas se evaluarán mediante dos pruebas escritas, una en el primer cuatrimestre y otra en el segundo, resultando como nota final de esta parte la media ponderada de ambas pruebas.
 - Las Actividades Prácticas y de Desarrollo se evaluarán mediante una única prueba que se realizará al final del curso.
 - Las Actividades Académicas Dirigidas se evalúan de forma continua durante el curso, teniendo en cuenta la asistencia, participación y corrección de los ejercicios propuestos. La nota final de esta parte será la media ponderada obtenida en las distintas sesiones.
- Para la realización de cualquier prueba evaluable no se permitirá el uso o consulta de documentación, salvo indicación expresa del profesor en convocatoria oficial.

Criterios de evaluación y calificación: (referidos a las competencias trabajadas durante el curso)

La nota oscilará entre 0 y 10 puntos, los cuales se acumularán en función de los porcentajes descritos a continuación:

- Enseñanzas básicas: 30%
- Actividades prácticas y de desarrollo: 45%
- Actividades académicas dirigidas: 25%

Para poder realizar la media ponderada el alumno deberá obtener al menos un punto sobre el cómputo global de 10 en cada uno de los módulos de enseñanza.

Las notas obtenidas en las pruebas de cualquiera de las tres partes no se guardarán para siguientes convocatorias.

Nota: Título II. Capítulo II. Artículo 14.2 y 14.3 de la Normativa de Régimen Académico y de Evaluación del Alumnado (aprobada en Consejo de Gobierno de la UPO el 18 de julio de 2006): “En la realización de trabajos, el **plagio** y la utilización de material no original, incluido aquél obtenido a través de Internet, sin indicación expresa de su procedencia y, si es el caso, permiso de su autor, podrá ser considerada causa de calificación de **suspense** de la asignatura, sin perjuicio de que pueda derivar en **sanción académica**.”

Corresponderá a la Dirección del Departamento responsable de la asignatura, oídos el profesorado responsable de la misma, los estudiantes afectados y cualquier otra instancia académica requerida por la Dirección del Departamento, decidir sobre la posibilidad de solicitar la apertura del correspondiente **expediente sancionador**”.

10. ORGANIZACIÓN DOCENTE SEMANAL. (Sólo hay que indicar el número de horas que a ese tipo de sesión va a dedicar el estudiante cada semana)								
SEMANA	Enseñanzas básicas (Gran Grupo) Nº de horas	Enseñanzas básicas y de desarrollo (Grupo de Docencia) Nº de horas (Indicación para un grupo*)	Actividades académicas dirigidas (Seminarios-Grupos de Trabajo) Nº de horas	Visita y excursiones Nº de horas	Tutorías especializadas Nº de horas	Control de lecturas obligatorias Nº de horas	Exámenes	Temas del temario a tratar
Primer Cuatrimestre								
SEMANA 1 (28 sep - 2 oct)	1,5							T1 (1.1-1.3)
SEMANA 2 (5-9 oct)	1,5							T1 (1.3-1.5)
SEMANA 3 (12-16 oct)	1,5							T2
SEMANA 4 (19-23 oct)	1,5	2						T3 (3.1-3.4) / S-APD1
SEMANA 5 (26-30 oct)	1,5							T3 (3.2-3.6)
SEMANA 6 (2-6 nov)	1,5	2						Clase de Repaso / S-APD2
SEMANA 7 (9-13 nov)								
SEMANA 8 (16-20 nov)	1,5	2						T4 / S-APD3
SEMANA 9 (23-27 nov)								
SEMANA 10 (30-4 dic)	1,5	2						T5 / S-APD4
SEMANA 11 (7,8-11 dic)	1,5							T5
SEMANA 12 (14-18 dic)								
SEMANA 13 (4,5,6-8 ene)								
SEMANA 14 (11-15 ene)		2	3					S-APD5 / S-AAD1
SEMANA 15 (18-22 ene)	1,5							T6
SEMANAS 16, 17 y 18 (25 ene – 12 feb) Evaluaciones finales							4	

* Esta columna indica las horas dedicadas a las APD por parte del grupo que primero recibe dicha clase, y por tanto marca el ritmo de avance en las APD. Para saber exactamente cuándo se imparte cada clase en un grupo determinados, se debe consultar el horario de APD correspondiente.

SEMANA	Enseñanzas básicas (Gran Grupo) Nº de horas	Enseñanzas básicas y de desarrollo (Grupo de Docencia) Nº de horas (Indicación para un grupo*)	Actividades académicas dirigidas (Seminarios-Grupos de Trabajo) Nº de horas	Visita y excursiones Nº de horas	Tutorías especializadas Nº de horas	Control de lecturas obligatorias Nº de horas	Exámenes	Temas del temario a tratar
Segundo Cuatrimestre								
SEMANA 1 (15-19 feb)								
SEMANA 2 (22-26 feb)	1,5							T7 (7.1-7.3)
SEMANA 3 (1-5 mar)	1,5	2						T7 (7.3-7.4) / S-APD6
SEMANA 4 (8-12 mar)	1,5		3					T7 (7.4-7.6) / S-AAD2
SEMANA 5 (15-19 mar)		2						S-APD7
SEMANA 6 (22-25 mar)	1,5							T8
SEMANA 7 (5 - 9 abr)		2	3					S-APD8 / S-AAD3
SEMANA 8 (12-16 abr)	1,5							T9(9.1)
SEMANA 9 (26-30 abr)	1,5	2						T9(9.2) / S-APD9
SEMANA 10 (3-7 may)								
SEMANA 11 (10-14 may)		2						S-APD10
SEMANA 12 (17-21 may)								
SEMANA 13 (24-28 may)								
SEMANA 14 (31 may-3,4 jun)								
SEMANA 15 (7-11 jun)			3					S-AAD4
SEMANAS 16-20 (14 jun – 14 jul) Evaluaciones finales							4	

* Esta columna indica las horas dedicadas a las APD por parte del grupo que primero recibe dicha clase, y por tanto marca el ritmo de avance en las APD. Para saber exactamente cuándo se imparte cada clase en un grupo determinados, se debe consultar el horario de APD correspondiente.

<p>11. TEMARIO DESARROLLADO. (Con indicación de las competencias que se van a trabajar en cada tema).</p>
<p align="center">BLOQUE 1: PROGRAMACIÓN ORIENTADA A OBJETOS EN JAVA</p>
<p>Tema 1: Introducción a la Orientación a Objetos en Java</p> <ol style="list-style-type: none"> 1. Generalidades de la Programación Orientada a Objetos 2. Introducción al Lenguaje Java 3. Tipos, operadores y expresiones 4. Estructura General de un programa en Java 5. Sentencias y bloques
<p>Tema 2: Interfaces</p> <ol style="list-style-type: none"> 1. Introducción 2. ¿Qué es una interfaz? 3. Definición de interfaces 4. Visibilidad 5. Herencia de Interfaces 6. Utilización de Interfaces 7. Ventajas de uso de las interfaces
<p>Tema 3: Clases: Implementación de Interfaces</p> <ol style="list-style-type: none"> 1. Introducción 2. Definición de Clases 3. Creación de Objetos 4. Atributos 5. Métodos 6. Encapsulación
<p>Tema 4: Reutilización y adaptación: Herencia vs. Composición</p> <ol style="list-style-type: none"> 1. Introducción 2. Adaptación por Composición 3. Adaptación por Herencia 4. Cuando usar Composición y Herencia
<p align="center">BLOQUE 2: ESTRUCTURAS DE DATOS</p>
<p>Tema 5: Clases de Fundamentales de Java (java.lang)</p> <ol style="list-style-type: none"> 1. Clase Object 2. Clase String 3. Clases Envoltura (en AAD) 4. Excepciones (en AAD)

Tema 6: Tablas 1. Introducción 2. Declaración 3. Acceso 4. Tablas multidimensionales 5. Inicialización de tablas 6. La clase java.util.Arrays
Tema 7: Introducción al Java Collection Framework 1. Introducción: Elementos del JCF 2. Interfaz Collection 3. Iteradores 4. Interfaces Comparable y Comparator 5. Utilidades para objetos contenedores: Clase java.util.Collections 6. Clase Arrays (ampliación)
Tema 8: Listas mediante la interfaz List 1. Introducción 2. Métodos 3. Implementación de List 4. Esquemas de recorrido
Tema 9: Conjuntos mediante las interfaces Set y SortedSet 1. Interfaz Set (Métodos y Clase HashSet) 2. Interfaz SortedSet (Métodos y Clase TreeSet)
<p style="text-align: center;">ACTIVIDADES PRÁCTICAS Y DE DESARROLLO</p> S-APD 1: Entorno de desarrollo java NetBeans S-APD 2: Estructuras de control y mecanismos de entrada/salida de texto S-APD 3: Interfaces e Introducción a las Clases S-APD 4: Ampliación de Clases S-APD 5: Herencia S-APD 6: Cadenas de Caracteres y Excepciones S-APD 7: Tablas S-APD 8: Colecciones e Iteradores S-APD 9: Listas, Comparadores y Clase Collections S-APD 10: Conjuntos y Conjuntos Ordenados
<p style="text-align: center;">ACTIVIDADES ACADÉMICAS DIRIGIDAS</p> S-AAD 1: Interfaces, Clases y Herencia S-AAD 2: Java Swing S-AAD 3: Tablas y Excepciones S-AAD 4: Java Collection Framework

12. MECANISMOS DE CONTROL Y SEGUIMIENTO. (Al margen de los contemplados a nivel general para toda la Experiencia Piloto, se recogerán aquí los mecanismos concretos que los docentes propongan para el seguimiento de cada asignatura).