

**Guía Docente de la Asignatura: Diseño y Programación Orientada a Objetos**

<b>Responsable</b>	Prof. D <sup>a</sup> . Luisa Sanz Martínez					
<b>Facultad</b>	Ciencias y Tecnología					
<b>Titulación</b>	Grado en Ingeniería Informática					
<b>Materia</b>	Informática					
<b>Plan</b>	2012					
<b>Carácter</b>	Formación Básica					
<b>Periodo de impartición</b>	Trimestral					
<b>Curso/es</b>	Primero					
<b>Nivel/Ciclo</b>	Grado					
<b>Créditos ECTS</b>	<b>Teóricos</b>	6	<b>Prácticos</b>	0	<b>Total</b>	6
<b>Lengua en la que se imparte</b>	Castellano					
<b>Datos de Contacto:</b>	Correo electrónico: luisa.sanz@ui1.es					

Asignaturas de la Materia	Asignaturas	Carácter	Curso	Créditos	Horas
	Estructura y tecnología de computadores I.	FB	1º	6	150
	Laboratorio de programación.	FB	1º	6	150
	Diseño y programación orientada a objetos.	FB	2º	6	150
	Estructura y tecnología de computadores II.	FB	1º	6	150
	Estructuras de datos y algoritmos.	FB	1º	6	150
Contextualización curricular de la asignatura	<p>La programación, en general, es la habilidad básica, fundamental e innegociable de cualquier Ingeniero Informático. Esta asignatura es la segunda etapa de la formación en programación, después de "Laboratorio de programación". Una vez aprendidas en "Laboratorio de programación" las estructuras esenciales de cualquier programa estructurado (variables, bucles, decisiones, funciones, etc.), en "Diseño y programación orientada a objetos" se aprenden los elementos del paradigma de la orientación a objetivos (clases, objetos, herencia, etc.), que es el más utilizado hoy en día, con mucha diferencia. Una vez terminada esta asignatura, el alumno tendrá los conocimientos y la capacidad necesarios para enfrentarse con cualquier problema real de programación. No obstante, para hacerlo con fundamentos científicos, siendo capaz de comparar cuantitativamente la bondad de varias soluciones alternativas a un mismo problema, el alumno también deberá cursar posteriormente "Estructura de datos y algoritmos".</p> <p>Los objetivos de la asignatura son:</p> <ul style="list-style-type: none"> <li>• Conocer los elementos del paradigma de la orientación a objetos</li> <li>• Reconocer, de forma básica, dichos elementos en un problema enunciado en lenguaje natural</li> <li>• Reforzar la habilidad de programación básica estructurada con ejercicios y prácticas con el lenguaje Java</li> <li>• Aprender el lenguaje de programación Java</li> <li>• Saber usar la ayuda centralizada de Java como ejemplo de búsqueda profesional de ayuda en problemas reales de programación</li> <li>• Introducir al alumno en el análisis y diseño orientado a objetos utilizando algunas estructuras básicas de UML</li> <li>• Conocer cómo conectar desde un programa Java a una base de datos relacional utilizando JDBC y SQL básico.</li> </ul>				
Prerrequisitos para cursar la asignatura	Ninguno.				

<p><b>Generales de la Materia</b></p>	<p>CB09: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.</p> <p>CB10: Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.</p>		
<p><b>Propias de la Universidad</b></p>	<p>CU09: Considerar los valores propios de la Formación Profesional Superior en términos de igualdad formativa y educativa con la universitaria.</p>	<p><b>Transversales</b></p>	<p>CT01: Capacidad de análisis y síntesis: encontrar, analizar, criticar (razonamiento crítico), relacionar, estructurar y sintetizar información proveniente de diversas fuentes, así como integrar ideas y conocimientos.</p> <p>CT04: Capacidad para la resolución de problemas.</p>
<p><b>Competencias de la Asignatura</b></p>	<p>CB09: Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería.</p> <p>CB10: Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas propios de la ingeniería.</p> <p>CT01: Capacidad de análisis y síntesis: encontrar, analizar, criticar (razonamiento crítico), relacionar, estructurar y sintetizar información proveniente de diversas fuentes, así como integrar ideas y conocimientos.</p> <p>CT04: Capacidad para la resolución de problemas.</p>		

<b>Actividades Formativas de la Materia</b>	<b>Trabajo dirigido</b>		<b>ECTS</b>	<b>HORAS</b>	<b>Trabajo autónomo del alumno</b>		<b>ECTS</b>	<b>HORAS</b>	
	<i>Comunidad de aprendizaje (Aula Virtual)</i>					Actividades de trabajo autónomo individual (Estudio de la Lección).	8	200	
	Actividades de descubrimiento inducido (Estudio del Caso).	5,76	154	Actividades de aplicación práctica (individuales).	2,88	72			
	Actividades de Interacción y colaboración (Foros-Debates de apoyo al caso y a la lección).	1,92	48	Lectura crítica, análisis e investigación.	3,6	90			
	Actividades de aplicación práctica (grupal online).	0	0	Actividades de evaluación.	0,4	10			
	Presentaciones de trabajos y ejercicios.	0	0	<i>Prácticas externas.</i>	0	0			
	Seminarios.	0	0	<i>Prácticas de iniciación profesional.</i>	0	0			
	<i>Interacción alumno-tutor (Aula Virtual).</i>					Trabajo Fin de Grado	0	0	
	Tutorías.	0,32	8						
	Presentaciones de trabajos y ejercicios propuestos.	0,64	16						
Actividades de evaluación.	0,48	12							
<b>Total</b>	<b>9,12</b>	<b>228</b>	<b>Total</b>	<b>14,88</b>	<b>372</b>				

  

Actividad	Descripción
<b>Trabajo dirigido.</b>	
<i>Comunidad de aprendizaje (Aula Virtual).</i>	
Actividades de descubrimiento inducido (Estudio del Caso).	Actividades en las que el alumno podrá llevar a cabo un aprendizaje contextualizado trabajando, en el Aula Virtual y de manera colaborativa, una situación real o simulada que le permitirá realizar un primer acercamiento a los diferentes temas de estudio.
Actividades de Interacción y colaboración (Foros-Debates de apoyo al caso y a la lección).	Actividades en las que se discutirá y argumentará acerca de diferentes temas relacionados con las asignaturas de cada materia y que servirán para guiar el proceso de descubrimiento inducido.
Actividades de aplicación práctica (grupal online).	Incluye la resolución de problemas, elaboración de proyectos y actividades similares que permitan aplicar los aspectos conceptuales, procedimentales y actitudinales trabajados en otras partes de las asignaturas.

Presentaciones de trabajos y ejercicios.	Incluye la elaboración conjunta en el Aula Virtual y, en su caso, defensa virtual de los trabajos y ejercicios solicitados conforme a los procedimientos de defensa que se establezcan en las guías docentes.
Seminarios.	Incluye la asistencia presencial o virtual a sesiones en pequeño grupo dedicadas a temáticas específicas de cada asignatura.
<i>Interacción alumno-tutor (Aula Virtual).</i>	
Tutorías.	Permiten la interacción directa entre docente y alumno para la resolución de dudas y el asesoramiento individualizado sobre distintos aspectos de las asignaturas.
Presentaciones de trabajos y ejercicios propuestos.	Incluye la elaboración individual, presentación y, en su caso, defensa virtual de los trabajos y ejercicios solicitados conforme a los procedimientos de defensa que se establezcan en las guías docentes.
Actividades de evaluación.	<i>Véase información al respecto en el apartado siguiente.</i>
<i>Trabajo Autónomo del alumno.</i>	
<i>Actividades de trabajo autónomo individual (Estudio de la Lección).</i>	Trabajo individual de los materiales utilizados en las asignaturas, aunque apoyado por la resolución de dudas y construcción de conocimiento a través de un foro habilitado para estos fines. Esta actividad será la base para el desarrollo de debates, resolución de problemas, etc.
Actividades de aplicación práctica (individuales).	Incluye el trabajo individual en la resolución de problemas, elaboración de proyectos y actividades similares que permitan aplicar los aspectos conceptuales, procedimentales y actitudinales trabajados en otras partes de la asignatura.
Lectura crítica, análisis e investigación.	Se trata de actividades en las que el alumno se acerca a los diferentes campos de estudio con una mirada crítica que le permite un acercamiento a la investigación. Se incluyen a modo de ejemplo, reseñas de libros o crítica de artículos y proyectos de investigación.
Actividades de evaluación.	<i>Véase información al respecto en el apartado siguiente.</i>

	Actividad	Descripción
<b>Actividades Formativas de la Asignatura</b>	<b>Trabajo dirigido.</b>	
	<i>Comunidad de aprendizaje (Aula Virtual).</i>	
	Actividades de descubrimiento inducido (Estudio del Caso).	Actividades en las que el alumno podrá llevar a cabo un aprendizaje contextualizado trabajando, en el Aula Virtual y de manera colaborativa, una situación real o simulada que le permitirá realizar un primer acercamiento a los diferentes temas de estudio.
	Actividades de Interacción y colaboración (Foros-Debates de apoyo al caso y a la lección).	Actividades en las que se discutirá y argumentará acerca de diferentes temas relacionados con las asignaturas de cada materia y que servirán para guiar el proceso de descubrimiento inducido.
	<i>Interacción alumno-tutor (Aula Virtual).</i>	
	Tutorías.	Permiten la interacción directa entre docente y alumno para la resolución de dudas y el asesoramiento individualizado sobre distintos aspectos de las asignaturas.
	<i>Trabajo Autónomo del alumno.</i>	
	<i>Actividades de trabajo autónomo individual (Estudio de la Lección).</i>	Trabajo individual de los materiales utilizados en las asignaturas, aunque apoyado por la resolución de dudas y construcción de conocimiento a través de un foro habilitado para estos fines. Esta actividad será la base para el desarrollo de debates, resolución de problemas, etc.
Actividades de aplicación práctica (individuales).	Incluye el trabajo individual en la resolución de problemas, elaboración de proyectos y actividades similares que permitan aplicar los aspectos conceptuales, procedimentales y actitudinales trabajados en otras partes de la asignatura.	
<b>Proceso de Aprendizaje</b>	<p>Cada unidad didáctica seguirá el siguiente esquema:</p> <p><b>Estudio de Caso de aplicación práctica:</b> se expondrá al alumno la problemática que originó las estructuras de programación que estudiará y practicará a lo largo de la unidad didáctica. Se pedirá que piense por sí mismo una solución de programación (o abstracta). Si es de programación, usará las estructuras de programación que actualmente ya sabe, o bien otras que tendrá que buscar por Internet. De este modo se verá totalmente inmerso en el tema y podrá comparar su solución con la aportada posteriormente en la unidad didáctica.</p> <p><b>Contenidos teóricos/Texto Canónico:</b> en los contenidos teóricos se expondrán las estructuras de programación, ejemplos resueltos (programas hechos) y numerosos ejercicios de programación que serán evaluados por el profesor.</p> <p><b>Foros de Debate:</b> cada alumno tendrá que programar individualmente un ejercicio que englobe todo lo aprendido en la unidad didáctica. Todos los alumnos tendrán que terminar el ejercicio en una fecha concreta. Una vez alcanzada dicha fecha, expondrá su solución en el foro al resto de compañeros, y estos criticarán la solución, aportarán mejoras, etc.</p>	

	<p><b>Trabajo Colaborativo/Webquest:</b> al igual que en el apartado anterior, cada alumno tendrá que programar individualmente otro ejercicio que englobe todo lo aprendido en la unidad didáctica, y presentarlo en el foro en una fecha concreta. Una vez realizado, buscará por Internet otras posibles soluciones al mismo problema, analizando ventajas e inconvenientes de ambas soluciones (la suya y la encontrada). El alumno no sólo hará esto con su propia solución, sino también con algunas soluciones de sus compañeros.</p>
<p><b>Orientaciones al estudio</b></p>	<p>Para aprender a programar, lo importante es, sencillamente, programar mucho y hacerlo de forma individual. Practicar, practicar y practicar. Igual que no se aprende a conducir un coche viendo conducir a tu profesor de autoescuela, es necesario que, para programar rápido y bien, realices individualmente numerosos ejercicios, cuantos más mejor. Incluso ejercicios similares. Por lo tanto el alumno, al estudiar un nuevo concepto, tendrá varios ejercicios a su disposición para practicar dicho concepto. No obstante, antes de lanzarse a realizar los ejercicios, es necesario que el alumno comprenda perfectamente los conceptos recién explicados y los relacione con los anteriores. Para ello puede preguntar directamente al profesor o bien utilizar el foro de la unidad didáctica para pedir consejo al resto de sus compañeros.</p>
<p><b>Resultados de Aprendizaje de la Materia</b></p>	<p>Al completar con éxito esta materia, el alumno:</p> <ul style="list-style-type: none"> <li>• Comprende el funcionamiento de un computador, y la necesidad de desarrollo de software por parte del programador.</li> <li>• Conoce los distintos paradigmas de programación, situando en ese contexto el lenguaje de programación que se va a utilizar.</li> <li>• Comprende la necesidad de un proceso de traducción de un lenguaje de alto nivel.</li> <li>• Conoce y distingue los conceptos de algoritmo y programa.</li> <li>• Conoce los tipos de datos primitivos y sus operaciones.</li> <li>• Distingue entre tipo de dato y objeto.</li> <li>• Conoce las acciones básicas de E/S de datos.</li> <li>• Aprende a usar las estructuras de control básicas: secuencial, condicional e iterativa.</li> <li>• Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado*.</li> <li>• Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos*.</li> <li>• Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje*.</li> <li>• Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos*.</li> <li>• Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases*.</li> <li>• Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos*.</li> <li>• Desarrolla programas, aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación*.</li> <li>• Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información*.</li> <li>• Gestiona información almacenada en bases de datos relacionales manteniendo la integridad y la consistencia de los datos*.</li> </ul>

- Comprende el origen de la orientación a objetos y sabe por qué se utilizan los objetos como clave para descomponer los sistemas en módulos en lugar de la funcionalidad.
- Conoce los principales conceptos de la tecnología de objetos y su aplicación en el diseño de software.
- Utiliza los principios de diseño de la orientación a objetos y aplicarlos en la resolución de problemas prácticos.
- Conoce las notaciones básicas de diseño que permiten representar las clases, sus relaciones y los objetos.
- Comprende el concepto de polimorfismo y de ligadura dinámica, y sabe aplicarlos en el diseño de clases.
- Conoce el concepto de herencia, sus distintas formas y los problemas originados en el diseño de lenguajes de programación.
- Comprende las relaciones complejas entre el sistema de tipos, la herencia y el polimorfismo.
- Posee habilidades para aplicar los conceptos de herencia, polimorfismo y ligadura dinámica para realizar diseños e implementaciones reutilizables.
- Sabe aplicar en la resolución de problemas concretos los principios de diseño con el objetivo de especificar jerarquías de clases y comprender los criterios para escoger entre una relación de herencia o de clientela.
- Caracteriza las instrucciones en lenguaje máquina y en lenguaje ensamblador. Distingue los diferentes formatos de las instrucciones y describe los diferentes métodos para codificar el código de operación.
- Distingue entre los diferentes tipos de instrucciones en ensamblador, modos de direccionamiento, registros, clases de arquitecturas a nivel de lenguaje máquina y tipos de operandos.
- Implementa un mismo programa en código ensamblador y en código de alto nivel.
- Explica cómo se implementan construcciones de los lenguajes de alto nivel en ensamblador y cómo se representan y almacenan en el computador datos y estructuras (vectores, matrices y registros).
- Describe una implementación elemental de camino de datos y unidad de control.
- Explica cómo la unidad de control de una CPU interpreta una instrucción a nivel máquina tanto en implementaciones cableadas como microprogramadas.
- Explica la estructura y el funcionamiento de la jerarquía de memoria en un computador y mostrar la necesidad de su presencia.
- Describe el hardware para gestión de la jerarquía de memoria en un computador (memoria cache, memoria virtual y protección). Describe el funcionamiento de la gestión de memoria virtual.
- Describe cómo configurar y diseñar memorias utilizando varios módulos. Explica cómo incrementar el ancho y número de palabras, junto con el diseño de memoria entrelazada.
- Describe las diferentes organizaciones de la memoria caché, analizando las posibles estrategias de extracción, colocación, reemplazo y actualización. Explica cómo se diseñaría una cache, analizando los parámetros que afectan a las prestaciones.
- Explica las diferentes técnicas de gestión de E/S. Describe controladores o interfaces de dispositivo.
- Explica el concepto de bus, estructuras y tipos. Describir los diferentes tipos de transferencia, el arbitraje, la temporización y el direccionamiento.
- Distingue entre arquitecturas CISC/RISC.
- Depura código a bajo nivel y desensambla.
- Reconoce la importancia de la abstracción y conoce los tipos de abstracciones que aparecen en programación: funcional, de datos, de iteradores y abstracción por generalización.
- Sabe diferenciar entre la especificación, representación e implementación de un tipo de dato abstracto, conociendo los conceptos de Función de Abstracción e Invariante de la Representación.



	<ul style="list-style-type: none"> <li>• Comprende cómo los conceptos de ocultamiento de información y encapsulamiento ayudan al desarrollo de tipos de datos más fiables.</li> <li>• Comprende los métodos de especificación: basados en una definición mediante axiomas o el método constructivo u operacional (basado en el uso de precondiciones y postcondiciones).</li> <li>• Es capaz de diseñar e implementar pequeñas aplicaciones para cada uno de los distintos tipos de datos que se imparten en la materia (listas, pilas, colas, colas con prioridad, conjuntos, diccionarios, árboles, tablas hash, grafos).</li> <li>• Adquiere la capacidad para comprender cómo el uso de distintos tipos de datos afecta a la eficiencia de los algoritmos que la usan.</li> <li>• Es capaz de implementar en lenguajes de alto nivel los tipos de datos propios de la materia así como otros definidos por el usuario.</li> <li>• Conoce las distintas representaciones e implementaciones de los tipos de datos que se imparten en la materia.</li> <li>• Es capaz de comparar implementaciones alternativas para un tipo de dato analizando los factores que influyen en la eficiencia y el uso de memoria.</li> <li>• Posee la capacidad de evaluar las necesidades de una aplicación específica, tomando decisiones justificadas sobre los tipos de datos y la representación más adecuada.</li> </ul>
<p><b>Resultados de Aprendizaje de la Asignatura</b></p>	<p>Al completar con éxito esta asignatura, el alumno:</p> <ul style="list-style-type: none"> <li>• Distingue entre tipo de dato y objeto.</li> <li>• Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos*.</li> <li>• Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje*.</li> <li>• Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos*.</li> <li>• Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases*.</li> <li>• Escribe programas que manipulen información, seleccionando y utilizando tipos avanzados de datos*.</li> <li>• Desarrolla programas, aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación*.</li> <li>• Comprende el origen de la orientación a objetos y sabe por qué se utilizan los objetos como clave para descomponer los sistemas en módulos en lugar de la funcionalidad.</li> <li>• Conoce los principales conceptos de la tecnología de objetos y su aplicación en el diseño de software.</li> <li>• Utiliza los principios de diseño de la orientación a objetos y aplicarlos en la resolución de problemas prácticos.</li> <li>• Conoce las notaciones básicas de diseño que permiten representar las clases, sus relaciones y los objetos.</li> <li>• Comprende el concepto de polimorfismo y de ligadura dinámica, y sabe aplicarlos en el diseño de clases.</li> <li>• Comprende las relaciones complejas entre el sistema de tipos, la herencia y el polimorfismo.</li> <li>• Posee habilidades para aplicar los conceptos de herencia, polimorfismo y ligadura dinámica para realizar diseños e implementaciones reutilizables.</li> <li>• Sabe aplicar en la resolución de problemas concretos los principios de diseño con el objetivo de especificar jerarquías de clases y comprender los criterios para escoger entre una relación de herencia o de clientela.</li> <li>• Comprende cómo los conceptos de ocultamiento de información y encapsulamiento ayudan al desarrollo de tipos de datos más fiables.</li> </ul>

## Plan de Evaluación

En el sistema de evaluación de la Universidad Internacional Isabel I de Castilla, en coherencia con la consecución gradual de competencias y resultados de aprendizaje que se ha descrito en la metodología, se dará preferencia a la evaluación continua complementada con una evaluación final presencial en cada unidad trimestral. Estas evaluaciones finales presenciales permiten obtener garantías respecto a la identidad del estudiante a la que se refiere la Guía de Apoyo para la elaboración de la Memoria de verificación de títulos oficiales universitarios (Grado y máster<sup>1</sup>) y a la veracidad del trabajo realizado durante el proceso de aprendizaje online, puesto que una parte importante de estas pruebas finales consiste en pruebas de verificación de la evaluación continua. Ésta será, por tanto, la vía preferente y recomendada por la Universidad para la obtención de los mejores resultados por parte del estudiante.

Sin embargo, es voluntad de esta Universidad ofrecer también una respuesta adecuada para aquellas personas que, por razones personales o profesionales, no pueden hacer un seguimiento de las asignaturas mediante el sistema de evaluación continua. No podemos olvidar que el perfil característico del estudiante de las universidades no presenciales se corresponde con personas de más de 25 años, en muchos casos con otros estudios universitarios y con responsabilidades profesionales y personales que deben compatibilizar con sus estudios online.

Teniendo en cuenta ambas perspectivas, el sistema de evaluación de la Universidad Internacional Isabel I de Castilla queda configurado de la siguiente manera:

- **Opción 1.** Evaluación continua más evaluación final. Los estudiantes que opten por esta vía podrán obtener hasta el 60% de la nota final a través de las actividades que se planteen en la evaluación continua. El 40% restante se podrá obtener en la prueba de evaluación final que se realizará de manera presencial. Esta prueba tendrá una parte dedicada a la verificación del trabajo realizado por el estudiante durante la evaluación continua (que se corresponde con el 60% de la nota final) y otra parte en la que realizarán diferentes pruebas teórico-prácticas para evaluar las competencias previstas en cada asignatura. La no superación de la parte de verificación implica que la calificación de la evaluación continua no se tendrá en cuenta y, por tanto, el 100% de la nota dependerá del resultado obtenido en la prueba final de evaluación de competencias.
- **Opción 2.** Evaluación final. Para los estudiantes que opten por esta vía, el 100% de la nota de la asignatura depende del resultado obtenido en esta prueba de evaluación final. Tanto en el proceso de información previa como en la formalización de la matrícula, el tutor informará de la existencia de esta posibilidad y valorará conjuntamente con cada persona su experiencia previa en la temática de la asignatura y otros factores que puedan influir en el resultado final.

Todos los estudiantes, independientemente de la opción seleccionada, tendrán derecho a una convocatoria extraordinaria de la prueba final de evaluación de competencias que se realizará después de finalizadas las pruebas de evaluación final ordinaria del conjunto de tres trimestres. Para los estudiantes de evaluación continua que no hayan superado la verificación y que también hayan suspendido la prueba de evaluación de competencias ordinaria, el 100% de la nota final dependerá del resultado obtenido en esta convocatoria extraordinaria o "Prueba de conjunto".

<sup>1</sup> Versión 0.1 - 22/03/2011 (Disponible en: [http://www.aneca.es/content/download/10717/120032/file/verifica\\_guia\\_11%324.pdf](http://www.aneca.es/content/download/10717/120032/file/verifica_guia_11%324.pdf))

Opciones	Seguimiento de la Evaluación Continua (EC)	Ponderación valor%		Opciones	Examen final de verificación de la EC	Examen final de validación de competencias	Total
Opción 1.	Si	60%	→	Opción 1.	Superado.	40%	100%
					No superado.	100%	100%
Opción 2.	No	0%	→	Opción 2.	No.	100%	100%

Tabla. Sistema de evaluación.

Nota: Si no se supera la *verificación* se pasa de la Opción 1 de evaluación a la Opción 2.

Los alumnos que no superen alguno/s de los exámenes finales trimestrales de validación de competencias pasarán a la evaluación extraordinaria que se celebrará un mes después de cada conjunto de tres trimestres y que se denominará "Prueba de conjunto".

Finalmente, las Prácticas externas y el Trabajo Fin de Grado (TFG) tendrán su propio sistema de evaluación, que se especificará en las Guías docentes correspondientes. El TFG, en todo caso, deberá ser defendido por el estudiante ante una Comisión de Evaluación.

El sistema de evaluación final será común para todas las asignaturas de la materia y se basará en una selección de las pruebas de evaluación más adecuadas para el tipo de competencias que se trabajen. Las pruebas de evaluación, on-line o presenciales, se clasifican de la siguiente forma (Montanero et al., 2006<sup>2</sup>):

1. Pruebas para evaluar competencias relacionadas con la comprensión, análisis, expresión de información (1, 2, 3, 4, 12).
2. Pruebas para evaluar competencias relacionadas con la aplicación de técnicas, procedimientos o protocolos de actuación y resolución de problemas (5, 6, 7, 13).
3. Pruebas para evaluar competencias relacionadas con la capacidad de investigar, pensar o actuar con creatividad y comunicarse verbalmente (8, 9, 12).
4. Pruebas para evaluar otras competencias profesionales, sociales y personales de carácter transversal (6, 9, 10, 11, 12).

<sup>2</sup> Montanero, M.; Mateos, V. L.; Gómez, V.; Alejo, R.: Orientaciones para la elaboración del Plan Docente de una Asignatura. Guía extensa. Badajoz, Universidad de Extremadura, Servicio de Publicaciones. 2006

Estrategias Evaluativas	Componentes de las competencias		
	Saber Competencias técnicas	Saber Hacer Competencias metodológicas	Saber ser-estar Competencias sociales y personales
Pruebas objetivas (tipo test).	x		
Pruebas semiobjetivas (preguntas cortas).	x		
Pruebas de desarrollo.	x		
Entrevista oral (en determinadas áreas).	x		x
Solución de problemas.	x	x	
Análisis de casos o supuestos prácticos.	x	x	x
Registros de observación sistemática.	x		
Proyectos y trabajos.	x	x	x
Entrevista (tutoría ECTS).	x	x	x
Pruebas de ejecución.	x	x	x
Solución de problemas.	x	x	x
Prueba de evaluación presencial.	x	x	x
Otros.			

Tabla. Estrategias o procedimientos de evaluación.

Los procedimientos de evaluación, al igual que ocurre con las actividades, se integran en el Sistema de Garantía Interna de Calidad (SGIC) de esta Universidad, de manera que la información recogida en cada trimestre se tendrá en cuenta en posteriores implementaciones de las asignaturas. La información acerca de la evaluación formará parte del compromiso público de la Universidad Internacional Isabel I de Castilla con sus estudiantes, de manera que las Guías docentes proporcionarán la información precisa sobre cómo se va a realizar el seguimiento de su trabajo y en qué va a consistir el sistema de evaluación de cada asignatura.

El sistema de calificaciones previsto para esta titulación se ajusta al Real Decreto 1125/2003, de 5 de septiembre, por el que se establece el sistema europeo de créditos y el sistema de calificaciones en las titulaciones universitarias de carácter oficial y de validez en todo el territorio nacional, que en su artículo 5, respecto al Sistema de calificaciones establece lo siguiente:

	<ul style="list-style-type: none"> <li>• La obtención de los créditos correspondientes a una materia comportará haber superado los exámenes o pruebas de evaluación correspondientes.</li> <li>• El nivel de aprendizaje conseguido por los estudiantes se expresará con calificaciones numéricas, que se reflejarán en su expediente académico junto con el porcentaje de distribución de estas calificaciones, sobre el total de alumnos que hayan cursado los estudios de la titulación en cada curso académico.</li> <li>• La media del expediente académico de cada alumno será el resultado de la aplicación de la siguiente fórmula: suma de los créditos obtenidos por el alumno multiplicados cada uno de ellos por el valor de las calificaciones que correspondan, y dividida por el número de créditos totales obtenidos por el alumno.</li> <li>• Los resultados obtenidos por el alumno en cada una de las materias del plan de estudios se calificarán en función de la siguiente escala numérica de 0 a 10, con expresión de un decimal, a la que podrá añadirse su correspondiente calificación cualitativa: 0-4,9: Suspenso (SS). 5,0-6,9: Aprobado (AP).7,0 -8,9: Notable (NT).9,0 -10: Sobresaliente (SB).</li> <li>• Los créditos obtenidos por reconocimiento de créditos correspondientes a actividades formativas no integradas en el plan de estudios no serán calificados numéricamente ni computarán a efectos de cómputo de la media del expediente académico.</li> <li>• La mención de Matrícula de Honor podrá ser otorgada a alumnos que hayan obtenido una calificación igual o superior a 9.0. Su número no podrá exceder del cinco por ciento de los alumnos matriculados en una materia en el correspondiente curso académico, salvo que el número de alumnos matriculados sea inferior a 20, en cuyo caso se podrá conceder una sola Matrícula de Honor.</li> </ul>
<p><b>Sistema de Calificación</b></p>	<p>Ponderación de la Evaluación Continua dentro del Proceso: 60%</p> <ul style="list-style-type: none"> <li>• Estudio de Caso Real de aplicación práctica: 10%</li> <li>• Contenidos teóricos/Texto Canónico: 20%</li> <li>• Foros de Debate: 15%</li> <li>• Trabajo Colaborativo/WebQuest: 15%</li> </ul> <p>Ponderación de la Evaluación Final dentro del Proceso: 40%</p> <ul style="list-style-type: none"> <li>• Prueba de Contenidos + Prueba de Validación del Alumno/a</li> </ul>

<p><b>Introducción</b></p>	<p>Los ejes temáticos de esta asignatura serán los siguientes:</p> <ul style="list-style-type: none"> <li>• Identificación de los elementos de un programa informático.</li> <li>• Utilización de objetos.</li> <li>• Uso de estructuras de control.</li> <li>• Desarrollo de clases.</li> <li>• Lectura y escritura de información.</li> <li>• Aplicación de las estructuras de almacenamiento.</li> <li>• Utilización avanzada de clases.</li> <li>• Mantenimiento de la persistencia de los objetos.</li> <li>• Gestión de bases de datos relacionales.</li> </ul>
<p><b>Breve Descripción de los Contenidos</b></p>	<p><b>UD1: Sintaxis básica del lenguaje Java</b></p> <ul style="list-style-type: none"> <li>• Características de Java: aquí relataremos el origen, la motivación y las principales características del lenguaje</li> <li>• Usar Netbeans: expondremos el principal entorno integrado de desarrollo de Java</li> <li>• Hola Mundo.java: utilizando Netbeans, realizaremos paso a paso un "Hola Mundo" (sencillo programa que imprime un mensaje por la pantalla). De este modo tendremos la ocasión de ver la estructura básica de un programa en Java</li> <li>• Comentarios: veremos todos los tipos de comentarios que existen en Java</li> <li>• Tipos de datos y variables: veremos las variables y tipos de datos básicos que existen</li> <li>• Expresiones: analizaremos cómo combinar variables, constantes y operaciones para formar expresiones más complejas</li> <li>• Entrada de datos por teclado: aprenderemos de forma básica a leer datos del teclado</li> <li>• Decisiones: veremos las expresiones booleanas, la estructura if y la estructura switch-case</li> <li>• Arrays: veremos las diferentes posibilidades que existen en Java para construir listas sencillas de elementos</li> <li>• Bucles: expondremos las estructuras que permiten construir bucles en Java</li> </ul> <p><b>UD2: Usar clases y objetos</b></p> <ul style="list-style-type: none"> <li>• Teoría de la orientación a objetos: aquí expondremos brevemente lo que son las clases, los objetos, los mensajes, el polimorfismo, las interfaces, la abstracción, la encapsulación y la herencia, utilizando ejemplos del mundo real. Insertaremos también cómo se pintan los diferentes conceptos enunciados en UML</li> <li>• Crear un objeto: veremos cómo crear un objeto a partir de una clase ya hecha</li> <li>• Usar un objeto: veremos cómo usar un objeto y como enviarle mensajes</li> <li>• Clases predefinidas: ¿cuáles son las principales clases predefinidas que nos aporta Java?</li> <li>• Arrays y objetos: descubriremos que un array es un tipo especial de objetos</li> <li>• El recolector de basura: enunciaremos el mecanismo que Java tiene para gestionar su memoria</li> <li>• Métodos y atributos estáticos: veremos lo que son y cómo utilizar los métodos y atributos estáticos de un objeto</li> <li>• Nomenclatura Java: expondremos la nomenclatura típica de Java y sus ventajas</li> <li>• Utilizar el API: nos acostumbraremos a utilizar el API público de Java y entender sus diferentes partes</li> </ul>

**UD3: Crear clases**

- Estructura de la clase: expondremos la estructura de una clase por dentro
- Declarar atributos: profundizaremos en la declaración de atributos, dentro de la estructura de la clase
- Escribir métodos: veremos cómo declarar y escribir métodos
- Usar NetBeans: ¿tiene Netbeans alguna característica que nos permita acelerar la escritura de las clases?
- Atributos y métodos estáticos: veremos cómo escribir métodos y atributos estáticos
- Paquetes: enunciaremos cómo agrupar las clases en estructuras superiores
- Encapsulación: cómo implementar la encapsulación en Java, con sus diferentes características
- Clase HolaMundo: ¡por fin entendemos en su totalidad la clase HolaMundo! Repasaremos y enunciaremos todas sus características
- Sobrecarga de métodos: cómo escribir varios métodos con el mismo nombre
- Constructores: qué son, para qué se utilizan, y cómo escribirlos
- Interfaces: repasaremos lo que son y cómo se desarrollan en Java

**UD4: Herencia**

- Herencia en Java: recordatorio del concepto y sintaxis básica que implementa la herencia en Java
- Herencia y atributos: ¿se heredan los atributos? Características y utilidad de este concepto.
- Herencia y métodos: ¿se heredan los métodos? Características y utilidad de este concepto.
- Clases y métodos finales: expondremos lo que son las clases y métodos finales, su sintaxis y su utilidad
- Clases y métodos abstractos: veremos las clases y métodos abstractos y su utilidad
- Ligadura dinámica: enunciaremos el importante concepto de ligadura dinámica, su utilidad y su desarrollo con Java

**UD5: Excepciones**

- Definición y utilidad: aquí expondremos para qué sirven y los problemas que solucionan
- Crear y lanzar una excepción: enunciaremos la sintaxis Java para crear y lanzar una excepción
- Capturar una excepción: veremos cómo capturar una excepción, con sus diferentes posibilidades
- Especificar excepciones: explicaremos cómo especificar una excepción para ser capturada posteriormente en otro método
- Excepciones Runtime: comprenderemos este tipo de excepciones y su utilidad
- Jerarquía de excepciones: volveremos a realizar un repaso al API, pero esta vez comprendiendo el concepto de excepciones y viendo la principal jerarquía de excepciones que existe

**UD6: Serialización y persistencia**

- Las clases de entrada/salida: enunciaremos las principales clases de entrada/salida y veremos en profundidad sus características
- Lectura/escritura de ficheros: veremos cómo leer y escribir en ficheros
- Serialización de objetos: veremos lo que es el estado de un objeto y cómo serializarlo para guardarlo o transmitirlo
- JDBC: nos introduciremos de forma básica en cómo recuperar y almacenar datos en una base de datos y relacionarlo con nuestros objetos

<p><b>Bibliografía Básica</b></p>	<p><b>Pérez, G.G. (2008). "Aprendiendo Java y POO".</b></p> <p>Este libro hace hincapié en la filosofía orientada a objetos más que en el lenguaje Java. Está pensado para que cualquier persona pueda aprender a programar con el paradigma de la orientación a objetos, incluso sin haber visto nunca un lenguaje estructurado o sin saber programar. Indispensable para aquéllos alumnos que no tienen clara la programación básica.</p> <p><b>Eckel, B. (2006). "Thinking in Java: The definitive introduction to object-oriented programming in the language of the world wide web".</b></p> <p>Con más de 1000 páginas, es la auténtica biblia del lenguaje Java. Explica lo que es la programación orientada a objetos utilizando Java como lenguaje práctico para implementar las ideas que se van desarrollando. La gran extensión del libro se debe a la gran cantidad de ejemplos resueltos (el código fuente ocupa mucho espacio). Por otra parte, el libro también desarrolla conceptos avanzados que no vemos en la asignatura, como concurrencia o programación gráfica.</p>
<p><b>Bibliografía Complementaria</b></p>	<p>Terrero, H. y Paredes, J. (2010). "Desarrollo de aplicaciones con Java". Ed. Fundación de Código Libre Dominicano.</p> <p>Jiménez, A. y Pérez, F.M. (2012). "Aprende a programar con Java. Un enfoque práctico partiendo de cero". Ed. Paraninfo.</p> <p>Schildt, H. (2012). "Java 7". Ed. Anaya.</p> <p>Sznajdleder, P. (2013). "Java a fondo: estudio del lenguaje y desarrollo de aplicaciones". Ed. Marcombo.</p> <p>Friesen, J. (2013). "Learn Java for Android development". Ed. Apress.</p> <p>Bloch, J. (2008). "Effective Java: a programming language guide". Ed. Addison Wesley.</p> <p>Sierra, K. (2005). "Head First Java (2nd edition)". Ed. O'Reilly.</p> <p>McLaughlin, B.D., Pollice, G. and West D. (2006). "Head first object-oriented analysis and design". Ed. O'Reilly.</p>
<p><b>Otros Recursos</b></p>	<p><a href="http://java.oracle.com">http://java.oracle.com</a> Web oficial de Java. Desde aquí se puede descargar todas las herramientas necesarias para la programación y ejecución de programas Java.</p> <p><a href="http://www.netbeans.org">www.netbeans.org</a> Web oficial del editor de desarrollo Netbeans.</p> <p><a href="http://docs.oracle.com/javase/tutorial/">http://docs.oracle.com/javase/tutorial/</a> Tutorial oficial de Java</p> <p><a href="http://es.wikipedia.org/wiki/Java_%28lenguaje_de_programaci%C3%B3n%29">http://es.wikipedia.org/wiki/Java_%28lenguaje_de_programaci%C3%B3n%29</a> Página de Wikipedia sobre el lenguaje Java. Muy útil para una visión general del lenguaje.</p> <p><a href="http://www.javahispano.org">www.javahispano.org</a> Estupendo portal sobre noticias, tutoriales y tecnologías Java</p> <p><a href="http://www.tecnun.es/asignaturas/Informat1/ayudainf/aprendainf/Java/Java2.pdf">http://www.tecnun.es/asignaturas/Informat1/ayudainf/aprendainf/Java/Java2.pdf</a> "Aprenda Java como si estuviera en primero". Famoso libro gratuito realizado por los profesores de la Universidad de Navarra y que explica Java de una forma directa y sencilla.</p>



<https://www.youtube.com/playlist?list=PL4D956E5314B9C253> Curso de 33 horas de Java en Youtube, utilizando Netbeans. Impresionante curso y muy ameno.

<http://java-source.net/> Sitio de recopilación de software libre para Java o relacionado con Java

<http://www.javaya.com.ar/> Tutorial para aprender Java sin saber absolutamente nada de programación

<http://www.desarrolloweb.com/manuales/57/> Otro estupendo tutorial para aprender Java

<http://codigofacilito.com/cursos/JAVA> Otro estupendo curso en vídeo de Java que también explica los conceptos de orientación a objetos.

<http://compunauta.com/forums/linux/programacion/java/ebook.html> Libro gratuito para aprender Java. Escrito magistralmente por un profesor de informática que quiso poner por escrito todos sus apuntes.

<http://www.etnassoft.com/biblioteca/introduccion-a-la-programacion-orientada-a-objetos-con-java/> Introducción a la programación orientada a objetos con Java. Libro gratuito.

<http://www.discoduroderoer.es/ejercicios-propuestos-y-resueltos-programacion-orientado-a-objetos-java/> Ejercicios resueltos de programación en Java.

<http://blog.fourthbit.com/2014/03/01/the-best-programming-language-or-how-to-stop-worrying-and-love-the-code/> Interesante comparativa sobre los lenguajes de programación más famosos. Un artículo imprescindible para tener una visión general de los lenguajes actuales de programación.

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html> Índice histórico de los lenguajes de programación más populares. Imprescindible para saber qué lenguajes de programación son los más usados. Java y C llevan años liderando el índice.