



GRADO

GUÍA DE ESTUDIO DE LA ASIGNATURA FUNDAMENTOS DE INFORMÁTICA

2ª PARTE | PLAN DE TRABAJO Y ORIENTACIONES PARA SU DESARROLLO



1.- PLAN DE TRABAJO

La asignatura de Fundamentos de la Informática requiere un estudio metódico y continuado a lo largo del cuatrimestre, dado que han de asimilarse conceptos teóricos y prácticos relacionados con la informática concentrados en una sola asignatura dentro de la carrera. Para facilitar su superación es conveniente planificar las etapas de estudio desde el principio, teniendo en cuenta los plazos de entrega y dedicando semanalmente el tiempo necesario, ya que es difícil asimilar la asignatura si se deja el trabajo para el final del cuatrimestre.

El objetivo de esta asignatura es que el estudiante adquiera los conocimientos y competencias reflejados en la guía general de la asignatura. Fundamentos de Informática tiene asignados 6 ETCS (créditos europeos), 4 de teoría y 2 de práctica. Un crédito equivale a 25 horas, lo que implica unas 150 horas de estudio y trabajo en total a lo largo de las 12 semanas disponibles para el curso.

Los créditos asignados están en consonancia con los contenidos, distribuidos en dos unidades didácticas de tres y nueve temas respectivamente, tal y como se detalla en el programa de la asignatura. En el caso de la segunda unidad didáctica, que cubre la mayor parte del curso, la organización de los estudios teóricos se ajusta al desarrollo de la práctica. Además, el libro base está orientado a la aplicación práctica de los conceptos durante todo el proceso de estudio. Por ello es importante abordar la parte teórica y práctica en paralelo siguiendo en la medida de lo posible el cronograma propuesto.

El cuadro siguiente muestra el cronograma que marca unas pautas adecuadas para que el alumno medio, que comienza a estudiar al principio del segundo semestre, alcance los objetivos al final del curso. Este cronograma incluye los contenidos de cada tema y las actividades a realizar, tanto de estudio teórico como práctico.

CONTENIDOS	PLAN DE ACTIVIDADES
UNIDAD DIDÁCTICA I: 1 SEMANA	
TEMA 1: Introducción , Semana 1	
<ol style="list-style-type: none"> 1. ¿qué es un sistema de computación? 2. Nociones básicas de hardware y software 	<ol style="list-style-type: none"> 1. Estudiar el documento con título "Introducción a la Informática", accesible a través de Curso virtual en el apartado del "Material de Estudio" de la "Unidad Didáctica I" 2. Realizar los ejercicios correspondientes del material de autoevaluación accesibles a través del Curso virtual en el apartado "Autoevaluación" de la "Unidad Didáctica I".
TEMA 2: Hardware. Semana 1	
<ol style="list-style-type: none"> 1. Sistema central. 2. Unidad Central de Proceso: reloj, unidad de control, unidad aritmético-lógica, registros, memoria central (características, tipos) 3. Unidades de Entrada/Salida. 4. Tipos (de comunicación, de almacenamiento) . 5. Controladores de periféricos. 6. Puertos. 7. Buses de interconexión. 8. El ordenador personal: evolución de los procesadores. 	<ol style="list-style-type: none"> 1. Estudiar los documentos accesibles a través de Curso virtual en el apartado "Tema 2. Hardware" del "Material de Estudio" de la "Unidad Didáctica I" 2. Realizar los ejercicios del material de autoevaluación accesibles a través del módulo de contenidos de Curso virtual en el apartado "Autoevaluación", de la Unidad Didáctica I.
TEMA 3: Fundamentos de sistemas operativos y redes . Semana 1	
<ol style="list-style-type: none"> 1. Sistema operativo <ol style="list-style-type: none"> 1. Definición y funciones de un sistema operativo: gestión de procesos, gestión de memoria, gestión de 	<ol style="list-style-type: none"> 1. Estudiar los documentos accesibles a través de Curso virtual en el apartado "Tema 3. " del "Material de Estudio" de la "Unidad Didáctica I"

<p>almacenamiento secundario, sistema de entrada/salida, gestión de ficheros, servicios de uso propio.</p> <ol style="list-style-type: none"> 2. Evolución de los sistemas operativos: Proceso por lotes, proceso interactivo, sistemas de tiempo compartido, Redes y sistemas multiprocesadores. 3. Arquitectura de un SO: procesador, planificador, gestor de archivos y de recursos, gestor de memoria, distribuidor. 4. Lenguaje de control (comandos). 5. Algunos sistemas operativos (ejemplos en el laboratorio de prácticas). 6. Programas del sistema: editores, debuggers, rutinas de utilidad... <ol style="list-style-type: none"> 2. Los lenguajes de programación <ol style="list-style-type: none"> 1. Tipos (maquina, bajo nivel, alto nivel) 2. Lenguajes de alto nivel: los diferentes paradigmas 3. Intérpretes y compiladores: Fases y tipos de errores. 	<ol style="list-style-type: none"> 2. Realizar los ejercicios correspondientes del material de autoevaluación accesibles a través del Curso virtual en el apartado "Autoevaluación" de la "Unidad Didáctica I". 3. Familiarizarse con el ordenador a nivel de usuario.
--	--

UNIDAD DIDÁCTICA II: 11 SEMANAS

TEMA 4: Objetos y clases. Semana 2

<ol style="list-style-type: none"> 1. Los conceptos de Objeto y Clase. 2. Creación de objetos. 3. Invocación de objetos.. 4. Parámetros. 5. Tipos de datos. 6. Instanciación de objetos. 7. Estado de un objeto.. 8. ¿Qué representa un objeto? 9. Interacción entre objetos. 	<ol style="list-style-type: none"> 1. Estudiar el capítulo 1 y Apéndice B del libro base de la "Unidad Didáctica II". 2. Estudiar los apuntes sobre clases y objetos, accesibles a través de Curso virtual en el apartado de la "Unidad Didáctica II" 3. Instalar el software BLUEJ y realizar los ejercicios sugeridos en el libro base. 4. Definir las clases necesarias en el problema de la práctica.
--	---

<ul style="list-style-type: none"> 10. Código fuente. 11. Valores de retorno. 12. Objetos como parámetros. 	
TEMA 5: Definición de clases. Semana 3	
<ul style="list-style-type: none"> 1. El concepto de Clase.. 2. Campos, constructores y métodos. 3. Paso de datos mediante parámetros. 4. Asignación de valores. 5. Tipos de métodos: <ul style="list-style-type: none"> 1. Métodos de acceso: get() 2. Métodos de modificación: set() 3. Método main() 4. Impresión desde métodos. 5. Estructuras de control: la sentencia condicional if. 6. Campos, parámetros y variables locales. 	<ul style="list-style-type: none"> 1. Estudiar el Capítulo 2 y leer los Apéndices B, C y D del libro base para la “Unidad Didáctica II”. 2. Leer los documentos accesibles a través de Curso virtual en el apartado “Clases y Objetos” del “Material de Estudio” de la “Unidad Didáctica II 3. Realizar los ejercicios en el entorno BLUEJ sugeridos en el libro base.
TEMA 6: Interacción entre objetos. Semana 4	
<ul style="list-style-type: none"> 1. Abstracción . 2. Modularidad. 3. Comparación de diagramas de clases con diagramas de objetos. 4. Tipos primitivos y tipos objeto.. 5. Objetos que crean objetos. 6. Constructores múltiples. 7. Llamadas a métodos. <ul style="list-style-type: none"> 1. Llamadas a métodos internos. 2. Llamadas a métodos externos. 	<ul style="list-style-type: none"> 1. Estudiar el Capítulo 3 del libro base para la “Unidad Didáctica II”. 2. Leer los Apéndices B y F del libro base para la “Unidad Didáctica II”. Realizar los ejercicios propuestos en el libro base. 3. Definir los campos y métodos necesarios para la resolución de la práctica.

8. Referencia a parámetros del propio objeto: la palabra clave this.	
9. Depuración de código.	

TEMA 7: Estructuras de almacenamiento y agrupación de objetos. Semana 5

<ol style="list-style-type: none"> 1. Librerías de clases. 2. Clases genéricas. 3. Colecciones de tamaño flexible: ArrayList <ol style="list-style-type: none"> 1. Procesamiento de colecciones. 2. Estructuras de control: los bucles for-each y while. 3. Acceso mediante índices e iteradores 4. Colecciones de tamaño fijo: Array. <ol style="list-style-type: none"> 1. Creación y declaración de arrays. 2. Uso de arrays. 3. Estructuras de control: el bucle for. 	<ol style="list-style-type: none"> 1. Estudiar el Capítulo 4 del libro base para la “Unidad Didáctica II”. 2. Realizar los ejercicios sugeridos en el libro base. 3. Realizar problemas de años anteriores en los exámenes resueltos que pueden encontrarse en el Curso virtual, sin contar aquellos que incluyen herencia. Poner especial hincapié en los problemas con implementación de bucles sobre estructuras de almacenamiento. 4. Definición de estructuras de almacenamiento e implementación de los métodos y constructores necesarios para la realización de la práctica.
---	--

TEMA 8: Comportamiento avanzado con objetos.

Semana 6

<ol style="list-style-type: none"> 1. Documentación de las clases de una librería. 2. Los paquetes y la sentencia import. 3. Visibilidad <ol style="list-style-type: none"> 1. Ocultamiento de la información. 2. Métodos y campos públicos y privados. 4. Variables de clase y constantes <ol style="list-style-type: none"> 1. La palabra clave static. 2. Constantes. 	<ol style="list-style-type: none"> 1. Estudiar las secciones 5.1, 5.5, 5.11 y 5.13 del Capítulo 5 del libro base para la “Unidad Didáctica II”. 2. Leer el Apéndice I del libro base para la “Unidad Didáctica II”. 3. Realizar los ejercicios correspondientes del libro base. 4. A partir de los ejercicios de exámenes resueltos de años anteriores, analizar el uso de clases estáticas y atributos privados y públicos. 5. Incluir la documentación en el código de la práctica y encapsular los campos.
--	--

TEMA 9: Acoplamiento entre clases. Semana 7

<ol style="list-style-type: none">1. Acoplamiento y cohesión..2. Uso de la encapsulación para reducir el acoplamiento.3. Ejecutar una aplicación fuera de BlueJ.	<ol style="list-style-type: none">1. Estudiar las secciones de 6..3 a 6.6 y el Apéndice E del libro base para la "Unidad Didáctica II".2. Realizar los ejercicios correspondientes del libro base.3. Revisar el acoplamiento y cohesión de la solución propuesta en la práctica.
--	--

TEMA 10: Extensión de clases: Herencia. Semana 8, 9 , 10

<ol style="list-style-type: none">1. El uso de la herencia.2. Jerarquías de herencia.3. Herencia en Java.4. Herencia y derechos de acceso5. Herencia e inicialización.6. Reutilización de código por medio de la herencia.7. Subtipos<ol style="list-style-type: none">1. Subclases y subtipos.2. Subtipos y asignación.3. Subtipo y paso de parámetros.4. Variables polimórficas.5. Enmascaramiento de tipos.8. La clase Object.9. Tipos estáticos y dinámicos.10. Sobreescritura de métodos.11. Llamada a métodos con la palabra reservada super.12. Métodos polimórficos13. Acceso protegido	<ol style="list-style-type: none">1. Estudiar los Capítulos 8 y 9 del libro base para la "Unidad Didáctica II".2. Realizar los ejercicios correspondientes del libro base.3. Realizar los ejercicios resueltos en exámenes de años anteriores en los que se utilice la herencia.
---	--

TEMA 11: Manejo de errores u excepciones. Semana 11

<p>Principios del lanzamiento de excepciones.</p> <ol style="list-style-type: none">1. Lanzar una excepción.	<ol style="list-style-type: none">1. E2.
--	---

<p>Principios del lanzamiento de excepciones</p> <p>Lanzar una excepción</p> <p>Las Clases Exception.</p> <p>El efecto de una excepción.</p> <p>Excepciones no comprobadas.</p> <p>Impedir la creación de un objeto.</p> <p>2. Manejo de excepciones</p> <ol style="list-style-type: none"> 1. Excepciones comprobadas: la cláusula throws. 2. Captura de excepciones: la sentencia try. 3. Lanzamiento y comprobación de excepciones. 4. Propagación de excepciones. 5. La cláusula finally. <p>3. Definición de nuevas clases de excepciones</p>	<p>Estudiar el Capítulo 12 del libro base para la “Unidad Didáctica II”.</p> <p>Incorporar el control de errores y excepciones en la práctica.</p>
<p>TEMA 12:Pruebas. Semana 12</p>	
<ol style="list-style-type: none"> 1. Prueba y depuración. 2. Pruebas de unidad. 3. Pruebas automatizadas. 	<p>Estudiar el capítulo 7 del libro base para</p> <p>Estudiar el captiulo 7 del libro base. Realizar las pruebas.</p>

Entrega de la práctica completa

2.- ORIENTACIONES PARA EL ESTUDIO DE LOS CONTENIDOS

El temario se estructura en dos Unidades Didácticas y ha sido planteado de tal forma que el alumno pueda introducirse en los contenidos de la asignatura de una manera gradual, adquiriendo los conocimientos necesarios para comprender e interactuar con un sistema informático, con un enfoque basado en la práctica de los mismos.

2.1 Unidad didáctica I: hardware y Software

2.1.1. Introducción

La primera Unidad Didáctica se estructura en tres temas. Se inicia con una breve introducción a la informática, para a continuación presentar los conceptos de Hardware y Software.

El primer tema es introductorio y motiva al estudio de la asignatura e introduce los conceptos más básicos de la informática. En el segundo tema se establece una visión global del concepto que da título al mismo y de los componentes fundamentales que integran el Hardware de un sistema informático. En el tercer tema se tratarán las nociones fundamentales de sistemas operativos y redes. En esta etapa es conveniente que el alumno se familiarice con el uso del ordenador en sesiones prácticas.

2.1.2. Contextualización

Estos contenidos deberán ser los primeros que el alumno tendrá que estudiar, y en este orden, antes de abordar contenidos más relacionados con la programación y la resolución de problemas por medio de un lenguaje de programación.

Los objetivos generales de esta unidad didáctica son:

1. Repaso histórico acerca de la evolución de los medios de computación.
2. Introducir los conceptos de computación, informática, mostrar la arquitectura clásica de un computador y las bases de las redes de computadores.
3. Introducir los fundamentos de los sistemas operativos, programas del sistema, lenguajes de programación, compiladores e intérpretes, de cara sentar los conceptos básicos que ayudarán a una mejor comprensión de la unidad didáctica II, centrada en un ámbito más práctico.

2.1.3. Resultados de aprendizaje asociados a los contenidos

Tema 1:

1. Entender la motivación del estudio de la asignatura.
2. Asimilar los conceptos más básicos de la informática.

Tema 2:

1. Asimilar los conceptos básicos del hardware: tipos de computadores, estructura física de un computador, etc.
2. Ser capaz de estimar la potencia de un computador dadas sus características.
3. Ser capaz de identificar los puertos adecuados para su uso en una situación determinada.

Tema 3:

1. Asimilar las nociones fundamentales de Software y conceptos relacionados, con especial hincapié en los sistemas operativos.
2. Ser capaz de identificar cuándo un problema se debe al mal funcionamiento del sistema operativo.
3. Ser capaz de identificar el método de proceso más adecuado en un caso determinado.
4. Ser capaz de identificar el tipo de lenguaje de programación más adecuado en un caso determinado.

2.1.4. Bibliografía básica

Los contenidos correspondientes a la *Unidad Didáctica I*

Para la introducción y el tema 3 se ponen unos apuntes a disposición de los alumnos en el curso virtual, para el tema 2 se puede encontrar en el libro:

- Informática Fundamental. Jesús M^a Minguet, Tim Read. Centro de estudios Ramón Areces, S.A. 2004. Capítulos 2 y 3. (El capítulo 2 se puede encontrar en el entorno virtual de Curso virtual).
- En el capítulo 2 se detalla el concepto de Hardware de un computador electrónico haciendo un repaso por sus características principales, para que el lector adquiriera un conocimiento básico a la hora de poder manejar un computador personal.

2.1.5. Materiales complementarios.

Como ejercicios de autoevaluación se proponen cuestiones y soluciones a dichas cuestiones, accesibles a través del Curso virtual en el apartado "Autoevaluación", y dentro de los apartados "Material de Estudio" de la Unidad Didáctica I.

2.2 Unidad didáctica II

2.2.1. Introducción

La segunda unidad didáctica se dedica a los fundamentos de la programación de computadores, con el objetivo de que los alumnos adquieran la base necesaria para realizar programas sencillos con una metodología apropiada. En esta unidad se plantean prácticas de programación con el lenguaje Java. La realización de la práctica es la herramienta básica para la asimilación de conceptos. Esta segunda unidad tiene una duración aproximada de doce semanas.

Los contenidos de esta segunda unidad didáctica se pueden estructurar en cuatro partes que, a su vez, establecen el orden en el que deberían estudiarse los diferentes conceptos teóricos presentes en esta unidad.

En una primera parte, el objetivo será fijar los conceptos básicos de la programación en Java. Para ello, deberán estudiarse las características fundamentales de clase, atributo y método (constructores, métodos que ayudan a la encapsulación de la clase y métodos de visualización).

Esta etapa requiere haber estudiado los contenidos del programa de la Unidad Didáctica I y se corresponde con los temas 4, 5 y 6 del programa detallado de la asignatura.

En una segunda parte se estudiará cómo utiliza las diferentes estructuras de control que permitan controlar el flujo de ejecución de un programa con estructuras de almacenamiento estáticas y dinámicas. Para el estudio de estos contenidos se requiere haber estudiado al menos la primera parte del programa de esta Unidad Didáctica II y se corresponde con el tema 7 del programa detallado.

En una tercera parte estudian los conceptos de composición y herencia, que nos permiten, respectivamente, incluir referencias a objetos dentro de otros objetos y reutilizar una clase existente extendiendo su funcionalidad. Esta etapa requiere haber estudiado los contenidos de la primera parte de esta Unidad Didáctica II y se corresponde con los temas 8, 9 y 10 del programa detallado de la asignatura.

En la última parte de esta Unidad Didáctica II se estudiarán aspectos relativos al manejo de excepciones, cuyo fin es la detección y corrección de errores. Además, se tratará la realización de pruebas necesarias en todo proceso de desarrollo software.

2.2.2 Descripción detallada de los contenidos

A continuación hacemos una breve introducción a cada uno de los temas:

Tema 4:

Corresponde a la presentación de los conceptos generales asociados a la programación en lenguaje Java: los objetos y las clases. Define el marco específico de desarrollo del temario.

Tema 5:

Este tema tiene una duración de algo más de una semana. Una vez entendida la semántica de clases, objetos y métodos en el capítulo anterior, en este tema se analizan los componentes principales del código fuente de una clase: atributos (campos), constructores y métodos. Se aborda también la forma en que se introducen datos en un método (paso de parámetro) y el acceso o modificación de datos de un objeto. Por último, se comienzan a abordar las estructuras de control de flujo y aspectos del manejo de variables mínimos para la implementación de una clase en JAVA.

Tema 6:

Con una duración prevista de poco más de una semana este tema introduce algunas relaciones entre clases. Para ello, se describen los conceptos de tipos predefinidos (enteros, booleanos, etc) y tipos objeto predefinidos o definidos por el programador. Se introduce la palabra clave *this*; paralelamente, se profundiza en la definición de constructores y el uso de métodos, así como el uso del depurador para el proceso de construcción de un programa.

Tema 7:

Este tema dura dos semanas y trata las diferentes formas en las que se puede agrupar objetos y el modo en que se puede procesar estas colecciones mediante estructuras de control de flujo.

Tema 8:

Este tema dura una semana y en él se introducen algunos aspectos cruciales para la generación de código de calidad, como son la correcta documentación de las clases, la encapsulación mediante métodos y atributos (campos) públicos y privados, y la definición de constantes y clases estáticas.

Tema 9:

La duración de este tema es de una semana. En él se tratará la programación en JAVA desde el punto de vista de la arquitectura del sistema, introduciendo las herramientas necesarias para evitar el acoplamiento entre clases.

Tema 10:

Este tema está previsto para una semana. También desde el punto de vista de la arquitectura del sistema, en este tema se introducen las herramientas necesarias para asegurar la posibilidad de reutilizar el código implementado ante nuevas especificaciones del problema, centrándose en el concepto de herencia.

Tema 11:

Se presentan los principales aspectos del manejo de errores mediante el lanzamiento y captura de excepciones.

Tema 12:

Este tema está previsto para menos de una semana. Se presentan los aspectos fundamentales de la realización de pruebas necesarias en el proceso de desarrollo software.

2.2.3. Contextualización

En el contexto de la asignatura, los objetivos de esta unidad didáctica son:

1. Entender los aspectos centrales de la estructura de un programa.
2. Comprender el concepto del paradigma orientado a objetos (POO) y destacar las características de cada uno.
3. Profundizar en la POO como el paradigma más importante hoy en día, y destacar sus características, conceptos y mecanismos básicos: clases, objetos, relaciones entre clases ...
4. Presentar Java como el LPOO más utilizado e introducir al estudiante en los aspectos principales del lenguaje y en cómo programar sirviéndose de éstos.
5. Conocer la existencia de algoritmos, estructuras de datos, instrucciones y operadores.
6. Entender las orientaciones definidas en la ingeniería del software y conocer la guía de estilo propia del lenguaje Java.
7. Conocer en qué consisten los posibles errores que pueden existir.

2.2.4. Resultados de aprendizaje asociados a los contenidos

Los resultados de aprendizaje para cada uno de los temas de esta unidad didáctica son los siguientes:

Tema4:

1. Asimilar los conceptos generales asociados a la programación en lenguaje JAVA: los objetos y clases.
2. Ser capaz de instanciar objetos de una clase e invocar métodos mediante el interfaz BLUEJ.
3. Ser capaz de hacer pequeñas modificaciones en un código JAVA.

Tema 5:

1. Entender desde el código fuente los componentes de una clase: campos, constructores y métodos.

2. Manejar operadores de asignación y sentencias condicionales.
3. Ser capaz de implementar métodos sencillos.

Tema 6:

1. Manejar operadores lógicos y matemáticos sobre tipos primitivos.
2. Ser capaz de implementar la creación de un objeto desde otra clase y llamar a sus métodos.
3. Ser capaz De realizar un diagrama de clases sencillo.
4. Ser capaz de hacer uso de un depurador en un entorno JAVA.

Tema 7:

1. Ser capaz de implementar en JAVA un recorrido sobre las estructuras de almacenamiento.
2. Entender los conceptos de biblioteca, paquete y objeto anónimo.

Tema 8:

1. Ser capaz de documentar una clase correctamente usando JavaDoc como estándar.
2. Ser capaz de distinguir entre aquellos atributos (campos) que deban ser privados o públicos, dado un problema y cuando una clase deba definirse como estática.

Tema 9: Ser capaz de definir un esquema de clases dado un problema, evitando el acoplamiento y la duplicación de código.

Tema 10:

1. Ser capaz de definir las relaciones de herencia entre clases dado un problema.
2. Ser capaz de extender una clase de una librería para crear una clase nueva.

Tema 11: Ser capaz de implementar una excepción sencilla en código JAVA.

Tema 12: Ser capaz de hacer un seguimiento manual de un programa e implementar sentencias de impresión.

2.2.5. Bibliografía básica

Los contenidos correspondientes a la *Unidad Didáctica II* se pueden encontrar en el libro:

▪Programación orientada a objetos con Java. Una introducción práctica usando BlueJ. 5 edición. David J. Barnes y Michael Kölling. Pearson / Prentice Hall. 2013. (Los capítulos que hay que estudiar se detallan en esta guía de estudio).

2.2.6. Material complementario

Como material complementario de la Unidad Didáctica II el equipo docente propone los siguientes documentos:

- “Vademecum Java”: guía de consulta rápida a la programación en Java (elaborado por el profesor José A. Mañas, profesor de la Universidad Politécnica de Madrid)
- Apuntes sobre Tipos Enumerados en Java.
- Apuntes sobre Colecciones en Java.

El acceso y/o versiones electrónicas de estos documentos puede encontrarse en el Curso virtual.

Otra referencia que pueden consultar es:

▪Java 2. J. Sánchez, G. Huecas, B. Fernández y P. Moreno. Osborne McGrawHill, 2005. (Los temas que hay que estudiar van desde las unidades 1 a la 6.)

2.2.7. Orientaciones concretas para el estudio de los contenidos

Tanto si el alumno está ya familiarizado con la programación, como si es la primera vez que se enfrenta a ella, se recomienda hacer hincapié en el diseño de la solución, haciendo uso de conceptos como objetos, herencia, composición o encapsulación, en los que se fundamenta la programación orientada a objetos.

3.- ORIENTACIONES PARA LA REALIZACIÓN DEL PLAN DE ACTIVIDADES

Un aspecto importante de la metodología es aplicar los conocimientos adquiridos por medio de los ejercicios propuestos en el libro base a realizar sobre un entorno JAVA, y la realización de forma progresiva de la práctica a medida que se vayan introduciendo los conceptos teóricos. Aunque los ejercicios del libro no están resueltos, el alumno puede solicitar ayuda al equipo docente a través de los canales establecidos en el curso virtual, y detallados en la guía del curso.

3.1. Medios y recursos

Los medios y recursos de los que dispone el alumno están descritos en la primera parte de la guía del curso (secciones 11 y 12).

3.2. Evaluación

La evaluación de la asignatura se llevará a cabo a partir de las siguientes pruebas:

- Realización de una práctica obligatoria a lo largo del cuatrimestre.
- Realización de un examen teórico/práctico.

3.2.1. Práctica de la Asignatura

El trabajo del curso incluye la realización de una práctica **obligatoria** de programación. El enunciado de la misma estará disponible tanto en el curso virtual de la asignatura.

Las prácticas son **corregidas por los Tutores de los Centros Asociados**, quienes las reenvían después a la sede central. La nota asignada por el tutor podrá incrementar **hasta un máximo de 1 punto** en la nota

final de la asignatura, por supuesto, siempre que esté aprobada. Las notas de las prácticas **no** se guardan de un curso para otro.

Las prácticas, por tanto, **se organizan en los centros asociados** bajo la responsabilidad de cada tutor, por lo que los alumnos deben ponerse en contacto con ellos **lo antes posible** al comienzo del curso para conocer:

1. El calendario de entrega y sesiones presenciales de las prácticas (tanto en **Junio** como en **Septiembre**)
2. La forma de entrega (correo electrónico, disquete, CD, curso virtual, etc)
3. La posibilidad de realizar sesiones especiales de tutoría de cara a la convocatoria de septiembre, etc.

El programa editor recomendado es BlueJ, conjuntamente con el compilador incluido en el JDK. Esto quiere decir que los tutores encargados de las clases de prácticas sólo están obligados a dar soporte para un entorno de esas características. Una versión para instalar y un pequeño manual de instalación se encuentran disponibles tanto en la página web de la asignatura como en el entorno virtual. Por tanto el uso de otros entornos no garantiza al alumno ningún tipo de soporte en la instalación, configuración o ejecución tanto del entorno como de los programas desarrollados.

Por otro lado, el uso de una versión u otra del JDK puede afectar a la compilación de los programas, ya que de una versión a otra se incluyen novedades, por lo que es importante especificar la versión del JDK que se usó para la realización de la práctica, o usar la especificada por el equipo docente en caso de que la haya.

Por último, los programas deben compilar independientemente del entorno de edición que haya sido usado, debiendo el alumno especificar claramente los pasos a seguir para el correcto funcionamiento de las aplicaciones generadas en las prácticas. **De no cumplirse este requisito y en caso de no poder ejecutar una práctica, ésta se considerará suspensa. Cualquier copia en las prácticas dará lugar a un suspenso para el todo el curso académico.**

3.2.2 Examen

El examen constará de dos partes, una teórica formada por preguntas tipo test y que será eliminatoria, y una segunda parte práctica formada por un problema de programación con varios apartados y en los que el alumno demostrará el nivel de los conocimientos adquiridos. Se incluirán también preguntas sobre la práctica obligatoria.

Para que el examen de un alumno sea calificado **deberá haber asistido, como mínimo, a tres sesiones presenciales de prácticas** en su centro asociado y haber entregado y aprobado la práctica obligatoria.

Para que un alumno pueda aprobar la asignatura deberá haber superado un mínimo de preguntas establecido en la parte teórica (tipo test) del examen.

En la evaluación de la asignatura se tendrán en cuenta especialmente los aspectos relativos al diseño más que a los detalles propios de la implementación.

4.- GLOSARIO

Puede encontrarse un glosario de términos relevantes para el estudio de la asignatura en el índice analítico del libro de referencia de la Unidad Didáctica II:

▪ *Programación orientada a objetos con Java. Una introducción práctica usando BlueJ.* David J. Barnes y Michael Kölling. Pearson / Prentice Hall. 2013.