



POLITÉCNICA

CAMPUS
DE EXCELENCIA
INTERNACIONAL

PROCESO DE
COORDINACIÓN DE LAS
ENSEÑANZAS PR/CL/001



E.T.S. de Ingenieros
Informaticos

ANX-PR/CL/001-01

GUÍA DE APRENDIZAJE

ASIGNATURA

105000110 - Programacion II

PLAN DE ESTUDIOS

10ML - Grado En Matematicas E Informática

CURSO ACADÉMICO Y SEMESTRE

2018/19 - Segundo semestre

Índice

Guía de Aprendizaje

1. Datos descriptivos.....	1
2. Profesorado.....	1
3. Conocimientos previos recomendados.....	3
4. Competencias y resultados de aprendizaje.....	3
5. Descripción de la asignatura y temario.....	4
6. Cronograma.....	6
7. Actividades y criterios de evaluación.....	9
8. Recursos didácticos.....	17
9. Otra información.....	18

1. Datos descriptivos

1.1. Datos de la asignatura

Nombre de la asignatura	105000110 - Programacion II
No de créditos	6 ECTS
Carácter	Basica
Curso	Primer curso
Semestre	Segundo semestre
Período de impartición	Febrero-Junio
Idioma de impartición	Castellano
Titulación	10ML - Grado en matematicas e informática
Centro en el que se imparte	10 - Escuela Tecnica Superior de Ingenieros Informaticos
Curso académico	2018-19

2. Profesorado

2.1. Profesorado implicado en la docencia

Nombre	Despacho	Correo electrónico	Horario de tutorías *
Angel Herranz Nieva	2309	angel.herranz@upm.es	L - 13:00 - 15:00 M - 10:00 - 12:00 M - 15:00 - 17:00
Raul Alonso Calvo	2307	raul.alonso@upm.es	L - 10:00 - 13:00 X - 10:00 - 13:00

Angel Lucas Gonzalez Martinez	D2304/1004B	lucas.gmartinez@upm.es	L - 16:30 - 18:00 M - 11:30 - 14:00 X - 11:30 - 13:30 Se debe concertar cita previa por email, con el fin indicar en que despacho estará el profesor
Jaime Ramirez Rodriguez (Coordinador/a)	5112	jaime.ramirez@upm.es	M - 11:00 - 14:00 J - 11:00 - 14:00
Santiago Eibe Garcia	2311	santiago.eibe@upm.es	L - 12:00 - 13:30 L - 15:30 - 17:00 X - 15:00 - 16:30 J - 15:30 - 17:00
Javier Galve Frances	2307	javier.galve@upm.es	M - 11:00 - 13:00 X - 11:00 - 14:00 J - 16:00 - 17:00
Jose Manuel Burgos Ortiz	2312	josemanuel.burgos@upm.es	M - 11:00 - 13:00 X - 11:00 - 13:00 J - 11:00 - 13:00

* Las horas de tutoría son orientativas y pueden sufrir modificaciones. Se deberá confirmar los horarios de tutorías con el profesorado.

3. Conocimientos previos recomendados

3.1. Asignaturas previas que se recomienda haber cursado

- Programacion I

3.2. Otros conocimientos previos recomendados para cursar la asignatura

- Leyes de De Morgan y álgebra de Boole
- Concepto de variables y concepto de ámbito de una variable
- Concepto de literal
- Manejo de arrays
- Bucles for, while y do while, así como sentencias condicionales if, if ... else, switch
- Concepto de función y procedimiento
- Operadores lógicos

4. Competencias y resultados de aprendizaje

4.1. Competencias

CE07 - Conocer los cimientos esenciales y fundacionales de la informática, subrayando los aspectos esenciales de la disciplina que permanecen inalterables ante el cambio tecnológico.

CE09 - Capacidad de elegir y usar los métodos analíticos y de modelización relevantes, y de describir una solución de forma abstracta.

CE11 - Comprender intelectualmente el papel central que tienen los algoritmos y las estructuras de datos, así como una apreciación del mismo.

CE13 - Poseer destrezas fundamentales de la programación que permitan la implementación de los algoritmos y las estructuras de datos en el software.

CE14 - Poseer las destrezas que se requieren para diseñar e implementar unidades estructurales mayores que

utilizan los algoritmos y las estructuras de datos, así como las interfaces por las que se comunican estas unidades.

CE43 - Capacidad para trabajar de forma efectiva como individuo, organizando y planificando su propio trabajo, de forma independiente o como miembro de un equipo.

CG01 - Capacidad de resolución de problemas aplicando conocimientos de matemáticas, ciencias e ingeniería.

CG02 - Capacidad para el aprendizaje autónomo y la actualización de conocimientos, y reconocimiento de su necesidad en las áreas de la matemática y la informática.

CG03 - Saber trabajar en situaciones carentes de información y bajo presión, teniendo nuevas ideas, siendo creativo.

CG05 - Capacidad de abstracción, análisis y síntesis.

CG08 - Capacidad de comunicarse de forma efectiva con los compañeros, usuarios (potenciales) y el público en general acerca de cuestiones reales y problemas relacionados con la especialización elegida.

CG10 - Capacidad para usar las tecnologías de la información y la comunicación.

4.2. Resultados del aprendizaje

RA58 - RA62 -

RA9 - Saber resolver de problemas aplicando conocimientos de matemáticas, ciencias e ingeniería.

5. Descripción de la asignatura y temario

5.1. Descripción de la asignatura

Esta asignatura se imparte en varios grados, pero es la misma asignatura y sólo varían los identificadores de las competencias. La versión aplicable durante el curso será la que esté publicada en moodle. Esa será la versión más actualizada de la guía y aplicable para todos los grados en los que se imparta la asignatura de programación 2.

Esta asignatura cubre los principios de la programación orientada a objetos, la implementación de estructuras de datos dinámicas basadas en cadenas enlazadas y el manejo de referencias. Asimismo, se introduce el concepto de Tipo Abstracto de Datos y se aplica en la resolución de problemas en los que es conveniente abstraerse de los detalles de implementación de las librerías empleadas.

Para poder abordar con éxito los conceptos que cubre esta asignatura es necesario haber asimilado los conceptos de la asignatura pre-requisito, programación 1.

Como lenguaje de programación para la realización de ejercicios y prácticas se usará el lenguaje de programación java en su versión 8 (jdk 1.8).

5.2. Temario de la asignatura

1. Clases y Objetos

1.1. De la programación imperativa a la programación orientada a objetos (POO)

1.2. Concepto de clase y Objeto

1.3. Concepto de referencia: variables de tipos básicos Vs referencias

2. Colecciones acotadas de objetos

3. Programación modular

3.1. Paquetes

3.2. Visibilidad

3.3. Automatización del proceso de pruebas unitarias: JUnit

4. Introducción al concepto de Tipo Abstracto de Datos (TAD) y sus aplicaciones

4.1. Concepto de TAD

4.2. Resolución de problemas con TADs

5. Herencia y Polimorfismo

5.1. Herencia Simple

5.2. Sobrescritura de métodos

5.3. Clases abstractas

5.4. Interfaces

6. Excepciones

7. Implementación de TADs lineales

7.1. Implementación de clases genéricas

7.2. Manejo de Cadenas enlazadas

7.3. Casos de Estudio: TAD Lista, TAD Pila, TAD Cola, etc.

6. Cronograma

6.1. Cronograma de la asignatura *

Sem	Actividad presencial en aula	Actividad presencial en laboratorio	Otra actividad presencial	Actividades de evaluación
1	Introducción a la Programación Orientada a Objetos: Clases y Objetos Duración: 02:00 LM: Actividad del tipo Lección Magistral	Introducción a un entorno de programación integrado y primeros ejemplos Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		
2	Clases y Objetos Duración: 00:20 LM: Actividad del tipo Lección Magistral Clases y Objetos Duración: 00:20 LM: Actividad del tipo Lección Magistral	Definiendo la primera clase en Java Duración: 01:40 PL: Actividad del tipo Prácticas de Laboratorio Ejercicios y Ejemplos de Clases y Objetos Duración: 01:40 PL: Actividad del tipo Prácticas de Laboratorio		
3	Clases y Objetos Duración: 00:20 LM: Actividad del tipo Lección Magistral	Ejercicios y Ejemplos de Clases y Objetos Duración: 01:40 PL: Actividad del tipo Prácticas de Laboratorio		
4	Colecciones acotadas (arrays de objetos) Duración: 00:30 LM: Actividad del tipo Lección Magistral	Ejemplos y ejercicios sobre colecciones acotadas Duración: 01:30 PR: Actividad del tipo Clase de Problemas		
5	Colecciones acotadas (arrays de objetos) Duración: 00:30 LM: Actividad del tipo Lección Magistral Colecciones acotadas (arrays de objetos) Duración: 00:30 LM: Actividad del tipo Lección Magistral	Ejemplos y ejercicios sobre colecciones acotadas Duración: 01:30 PR: Actividad del tipo Clase de Problemas Ejemplos y ejercicios sobre colecciones acotadas Duración: 01:30 PR: Actividad del tipo Clase de Problemas		
6	Programación modular Duración: 01:30 LM: Actividad del tipo Lección Magistral	Ejemplos y ejercicios sobre colecciones acotadas Duración: 01:00 PR: Actividad del tipo Clase de Problemas Ejemplos prácticos sobre el uso y definición de módulos, paquetes y librerías Duración: 01:30 PL: Actividad del tipo Prácticas de Laboratorio		Examen tipo problema EX: Técnica del tipo Examen Escrito Evaluación continua Duración: 02:00

7	<p>Introducción al concepto de Tipo Abstracto de Datos (TAD) y sus aplicaciones</p> <p>Duración: 01:30 LM: Actividad del tipo Lección Magistral</p>	<p>Resolución de programas usando TADS/librerías/módulos</p> <p>Duración: 02:30 PL: Actividad del tipo Prácticas de Laboratorio</p>		
8		<p>Resolución de programas usando TADS/librerías/módulos</p> <p>Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio</p>		<p>Ejercicio evaluable (NE)</p> <p>OT: Otras técnicas evaluativas Evaluación continua Duración: 02:00</p>
9	<p>Concepto de herencia</p> <p>Duración: 00:20 LM: Actividad del tipo Lección Magistral</p> <p>Herencia: redefinición o sobrescritura de métodos</p> <p>Duración: 00:30 LM: Actividad del tipo Lección Magistral</p>	<p>Ejercicios sobre herencia simple</p> <p>Duración: 01:30 PR: Actividad del tipo Clase de Problemas</p> <p>Problemas sobre herencia y sobrescritura</p> <p>Duración: 01:30 PL: Actividad del tipo Prácticas de Laboratorio</p>	<p>Tutoría de la práctica obligatoria en grupo</p> <p>Duración: 02:00 OT: Otras actividades formativas</p> <p>Tutoría de la práctica obligatoria en grupo</p> <p>Duración: 02:00 OT: Otras actividades formativas</p>	<p>Proyecto de programación (NP). El coordinador de la asignatura indicará por medio del foro moodle de la asignatura los plazos para entregarla y cómo debe realizarse la entrega.</p> <p>OT: Otras técnicas evaluativas Evaluación continua y sólo prueba final Duración: 10:00</p>
10	<p>Herencia abstracta</p> <p>Duración: 01:00 LM: Actividad del tipo Lección Magistral</p> <p>Interfaces de programación</p> <p>Duración: 01:00 LM: Actividad del tipo Lección Magistral</p>	<p>Problemas sobre herencia</p> <p>Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio</p>	<p>Tutoría de la práctica obligatoria en grupo</p> <p>Duración: 02:00 OT: Otras actividades formativas</p>	<p>Proyecto de programación (NP). El coordinador de la asignatura indicará por medio del foro moodle de la asignatura los plazos para entregarla y cómo debe realizarse la entrega.</p> <p>OT: Otras técnicas evaluativas Evaluación continua y sólo prueba final Duración: 10:00</p>
11	<p>Concepto de excepción, manejo y creación de excepciones</p> <p>Duración: 00:20 LM: Actividad del tipo Lección Magistral</p>	<p>Ejemplos y ejercicios sobre el manejo de excepciones</p> <p>Duración: 00:40 PR: Actividad del tipo Clase de Problemas</p> <p>Ejemplos y ejercicios sobre definición de excepciones</p> <p>Duración: 01:00 PR: Actividad del tipo Clase de Problemas</p>	<p>Tutoría de la práctica obligatoria en grupo</p> <p>Duración: 02:00 OT: Otras actividades formativas</p>	<p>Proyecto de programación (NP). El coordinador de la asignatura indicará por medio del foro moodle de la asignatura los plazos para entregarla y cómo debe realizarse la entrega.</p> <p>OT: Otras técnicas evaluativas Evaluación continua y sólo prueba final Duración: 10:00</p>
12	<p>Implementación de genéricos</p> <p>Duración: 00:30 LM: Actividad del tipo Lección Magistral</p> <p>Implementación de TADs lineales</p> <p>Duración: 00:30 LM: Actividad del tipo Lección Magistral</p>	<p>Ejemplos y ejercicios de implementación de genéricos</p> <p>Duración: 01:30 PR: Actividad del tipo Clase de Problemas</p> <p>Ejemplos y ejercicios sobre implementación de TADs</p> <p>Duración: 01:30 PL: Actividad del tipo Prácticas de Laboratorio</p>		<p>Proyecto práctico de programación opcional (NPO)</p> <p>OT: Otras técnicas evaluativas Evaluación continua Duración: 10:00</p>
13		<p>Ejemplos y ejercicios sobre implementación de TADs</p> <p>Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio</p> <p>Ejemplos y ejercicios sobre implementación de TADs</p> <p>Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio</p>		<p>Proyecto práctico de programación opcional (NPO)</p> <p>OT: Otras técnicas evaluativas Evaluación continua Duración: 10:00</p>

14	Implementación de TADs lineales Duración: 00:30 LM: Actividad del tipo Lección Magistral	Ejemplos y ejercicios sobre implementación de TADs Duración: 01:30 PL: Actividad del tipo Prácticas de Laboratorio Ejemplos y ejercicios sobre implementación de TADs Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		Proyecto práctico de programación opcional (NPO) OT: Otras técnicas evaluativas Evaluación continua Duración: 10:00
15		Ejemplos y ejercicios sobre implementación de TADs Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio Ejemplos y ejercicios sobre implementación de TADs Duración: 02:00 PL: Actividad del tipo Prácticas de Laboratorio		Prueba individual de evaluación (NT) EX: Técnica del tipo Examen Escrito Evaluación continua Duración: 03:00
16				
17				Examen final (NT) EX: Técnica del tipo Examen Escrito Evaluación sólo prueba final Duración: 03:00

Las horas de actividades formativas no presenciales son aquellas que el estudiante debe dedicar al estudio o al trabajo personal.

Para el cálculo de los valores totales, se estima que por cada crédito ECTS el alumno dedicará dependiendo del plan de estudios, entre 26 y 27 horas de trabajo presencial y no presencial.

* El cronograma sigue una planificación teórica de la asignatura y puede sufrir modificaciones durante el curso.

7. Actividades y criterios de evaluación

7.1. Actividades de evaluación de la asignatura

7.1.1. Evaluación continua

Sem.	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
6	Examen tipo problema	EX: Técnica del tipo Examen Escrito	Presencial	02:00	15%	3 / 10	
8	Ejercicio evaluable (NE)	OT: Otras técnicas evaluativas	Presencial	02:00	15%	3 / 10	
9	Proyecto de programación (NP). El coordinador de la asignatura indicará por medio del foro moodle de la asignatura los plazos para entregarla y cómo debe realizarse la entrega.	OT: Otras técnicas evaluativas	No Presencial	10:00	10%	5 / 10	
10	Proyecto de programación (NP). El coordinador de la asignatura indicará por medio del foro moodle de la asignatura los plazos para entregarla y cómo debe realizarse la entrega.	OT: Otras técnicas evaluativas	No Presencial	10:00	10%	5 / 10	
11	Proyecto de programación (NP). El coordinador de la asignatura indicará por medio del foro moodle de la asignatura los plazos para entregarla y cómo debe realizarse la entrega.	OT: Otras técnicas evaluativas	No Presencial	10:00	10%	5 / 10	
12	Proyecto práctico de programación opcional (NPO)	OT: Otras técnicas evaluativas	No Presencial	10:00	3.33%	0 / 10	
13	Proyecto práctico de programación opcional (NPO)	OT: Otras técnicas evaluativas	No Presencial	10:00	3.33%	0 / 10	
14	Proyecto práctico de programación opcional (NPO)	OT: Otras técnicas evaluativas	No Presencial	10:00	3.33%	0 / 10	

15	Prueba individual de evaluación (NT)	EX: Técnica del tipo Examen Escrito	Presencial	03:00	40%	4 / 10	CG01 CG02 CG03 CG05 CG08 CG10 CE07 CE09 CE11 CE13 CE14 CE43
----	--------------------------------------	-------------------------------------	------------	-------	-----	--------	--

7.1.2. Evaluación sólo prueba final

Sem	Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
9	Proyecto de programación (NP). El coordinador de la asignatura indicará por medio del foro moodle de la asignatura los plazos para entregarla y cómo debe realizarse la entrega.	OT: Otras técnicas evaluativas	No Presencial	10:00	10%	5 / 10	
10	Proyecto de programación (NP). El coordinador de la asignatura indicará por medio del foro moodle de la asignatura los plazos para entregarla y cómo debe realizarse la entrega.	OT: Otras técnicas evaluativas	No Presencial	10:00	10%	5 / 10	
11	Proyecto de programación (NP). El coordinador de la asignatura indicará por medio del foro moodle de la asignatura los plazos para entregarla y cómo debe realizarse la entrega.	OT: Otras técnicas evaluativas	No Presencial	10:00	10%	5 / 10	
17	Examen final (NT)	EX: Técnica del tipo Examen Escrito	Presencial	03:00	70%	4 / 10	CG01 CG02 CG03 CG05 CG08 CG10 CE07 CE09 CE11 CE13 CE14 CE43

7.1.3. Evaluación convocatoria extraordinaria

Descripción	Modalidad	Tipo	Duración	Peso en la nota	Nota mínima	Competencias evaluadas
Examen final (véase Criterios de Evaluación)	EX: Técnica del tipo Examen Escrito	Presencial	03:00	70%	4 / 10	
Proyecto de programación (NP). El coordinador de la asignatura indicará por medio del foro de moodle los plazos para entregarla y como se debe realizarse la entrega.	OT: Otras técnicas evaluativas	Presencial	30:00	30%	5 / 10	CG01 CG02 CG03 CG05 CG08 CG10 CE07 CE09 CE11 CE13 CE14 CE43

7.2. Criterios de evaluación

Con el fin de superar esta asignatura, el alumno debe seguir uno de estos dos itinerarios alternativos: **basado en evaluación continua o basada en examen final. El alumno podrá elegir uno de estos dos itinerarios de la siguiente manera:**

- **La evaluación continua:** es la asignación por defecto para cualquier alumno.
- **La evaluación por prueba final:** en el periodo establecido por el coordinador y siguiendo las indicaciones establecidas por este. El alumno no podrá cambiar de itinerario posteriormente, salvo causa de fuerza mayor debidamente justificada.

Una vez que el alumno haya realizado todas las actividades evaluables del itinerario escogido, el alumno obtendrá su nota en la convocatoria ordinaria. Si suspende en la convocatoria ordinaria, dispondrá de la convocatoria extraordinaria.

El código fuente que entregue el alumno en los ejercicios y en las prácticas debe compilar en la **java**.

Normas sobre copias

Se aplicará la normativa vigente sobre copias en la UPM recogida en la normativa de exámenes publicada en la página web de la UPM.

Si además, los alumnos que han copiado la práctica o ejercicio reconocen haberlo hecho sin el consentimiento de los alumnos plagiados (apropiación indebida, hurto, robo, sustracción, etc.) o son reincidentes, se solicitará la apertura del correspondiente expediente académico.

Lo indicado anteriormente se aplicará sin menoscabo de los derechos y deberes de los estudiantes universitarios que están desarrollados en los Estatutos de la Universidad Politécnica de Madrid (BOCM de 15 de noviembre de 2010) y en el Estatuto del Estudiante Universitario (RD 1791/2010 de 30 de diciembre).

El artículo 124 a) de los estatutos de la UPM fija como deber del estudiante ... "Seguir con responsabilidad y aprovechamiento el proceso de formación, adquisición de conocimientos, y aprendizaje correspondiente a su condición de universitario "... y el artículo 13 del Estatuto del Estudiante Universitario, en el punto d) especifica también como deber del estudiante universitario "abstenerse de la utilización o cooperación en procedimientos fraudulentos en las pruebas de evaluación, en los trabajos que se realicen o en documentos oficiales de la universidad". En el caso de que en el desarrollo de las pruebas de evaluación se aprecie el incumplimiento de los deberes como estudiante universitario, el coordinador de la asignatura podrá ponerlo en conocimiento del Director o Decano del Centro, que de acuerdo con lo establecido en el artículo 74 (n) de los Estatutos de la UPM tiene competencias para "Proponer la iniciación del procedimiento disciplinario a cualquier miembro de la Escuela o Facultad, por propia iniciativa o a instancia de la Comisión de Gobierno" al Rector, en los términos previstos en los estatutos y normas de aplicación.

Normas sobre la realización de exámenes

El alumno que vaya a realizar alguno de los exámenes o pruebas de evaluación presencial tendrá que apuntarse en un formulario dispuesto a tal efecto cuando así sea requerido por el profesorado a través de moodle. Este formulario estará habilitado unos días antes del examen o prueba. **Si un alumno no se apunta y se presenta al examen o prueba, podría ser penalizado con 0.5 puntos sobre la nota total del examen. Si la aplicación de esta penalización implicara el suspenso de la asignatura, la penalización sería convenientemente rebajada hasta que esta circunstancia desapareciera.**

Sobre las notas de la práctica y los ejercicios evaluables de clase de los cursos 2017/2018 y anteriores

Estas notas se guardarán, salvo que el alumno realice alguna de las pruebas equivalentes recogidas en esta guía. En ese momento, el alumno perderá la nota guardada y se quedará con la nota que obtenga en la prueba del presente curso. La práctica equivale al proyecto y los ejercicios evaluables al ejercicio evaluable de este presente curso.

A continuación, se explican los criterios de evaluación de cada itinerario y los de la convocatoria extraordinaria:

1. Itinerario basado en evaluación continua

Si el alumno sigue este itinerario tendrá que realizar **actividades evaluables** de los siguientes tres tipos:

- **Pruebas de evaluación individual:** se realizarán durante el periodo lectivo. **Una se hará sobre la semana 5 o 6 con un peso del 15% (NT1) y otra sobre la semana 16 con un peso del 40% (NT2).** A partir de las calificaciones obtenidas en estas pruebas se obtendrá la **nota de teoría (NT)** de la asignatura, que será un valor numérico entre 0 y 10. **El alumno deberá obtener una nota mínima de 4 en esta parte para superar la asignatura.**
- **Ejercicio evaluable:** será un ejercicio que se realizará individualmente o en parejas en el horario de clase. Se entregarán por los medios telemáticos que se especifiquen en el horario de clase. A partir de las calificaciones obtenidas en estos ejercicios se obtendrá la **nota del ejercicio evaluable (NE)** de la asignatura, que será un valor numérico entre 0 y 10. **El alumno deberá obtener una nota mínima de 3 en esta parte para superar la asignatura.**
- **Proyecto de Programación:** serán ejercicios que se realizarán individualmente o en parejas **FUERA** del horario de clase. Se entregarán a través de un sistema de entrega web en los periodos que se establezcan. **El mecanismo de entrega será comunicado por el coordinador de la asignatura en el foro de moodle.** A partir de las calificaciones obtenidas en estos ejercicios se obtendrá la **nota del proyecto de programación (NP)** de la asignatura, que será un valor numérico entre 0 y 10. **El alumno deberá obtener una nota mínima de 5 en esta parte para superar la asignatura.**
- **Proyecto de programación opcional:** Sería un ejercicio práctico el cual se realizará bajo las condiciones que se establezcan en su momento. La calificación que se obtenga en este ejercicio será la **nota del proyecto de programación opcional (NPO)** de la asignatura, que será un valor numérico entre 0 y 10.

En la tabla de actividades evaluables se puede encontrar **una estimación aproximada** del momento en el que se realizará cada actividad y el peso que tendrá en la nota final de la asignatura.

La fórmula para calcular la **nota final de la asignatura (NF)** es la siguiente:

$$NF = NT + 0.30NP + 0.15NE + 0.10NPO, \text{ si } NT \geq 4, NP \geq 5 \text{ y } NE \geq 3$$

NF = 0, e.o.c.

Donde NT = Max (NT2*0.55, NT1*0.15+NT2*0.40) (Según esta fórmula una mala nota en el primer examen, será sustituida por la nota del segundo examen, si esta es más favorable)

Observaciones:

- Si un alumno obtiene una nota de teoría (NT) inferior a 4 en la convocatoria ordinaria, o una nota inferior a 5 en la nota de prácticas (NP) o una nota inferior a 3 en el ejercicio evaluable (NE), suspende la asignatura en la convocatoria ordinaria.
- Si un alumno suspende la asignatura en la convocatoria ordinaria, las NT, NP, NPO y la NE aprobadas (nota ≥ 5) se guardarán para la convocatoria extraordinaria.
- **Si un alumno es repetidor del curso 2017/2018 o anteriores** y tiene aprobadas las prácticas o los ejercicios evaluables, pero desea mejorar su nota en estas actividades evaluables, lo podría hacer realizando las actividades que se propongan. Sin embargo, en el momento que entregue el primer proyecto de programación o ejercicio evaluable perderá la NP o la NE respectivamente obtenidas en el anterior curso académico.

2. Itinerario basado en prueba final

Si el alumno sigue este itinerario tendrá que realizar **actividades evaluables** de los siguientes tipos:

- **Examen final:** será un examen que constará de dos partes: 1) un examen teórico (test + problema) cuya nota (NT) tendrá un peso de 55% en la nota de la asignatura; y 2) un ejercicio práctico cuya nota (NE) tendrá un peso de un 15% en la nota de la asignatura. Ambas notas se representarán con un valor numérico entre 0 y 10.
- **Proyecto de programación:** serán ejercicios que se realizarán individualmente o en parejas. Se entregarán a través de un sistema de entrega web en los periodos que se establezcan. **El mecanismo de entrega será comunicado por el coordinador de la asignatura en el foro de moodle.** A partir de las calificaciones obtenidas en estos ejercicios se obtendrá la **nota de prácticas (NP)** de la asignatura, que será un valor numérico entre 0 y 10.

En la tabla de actividades evaluables se puede encontrar una estimación aproximada del momento en el que se realizará cada actividad y el peso que tendrá en la nota final de la asignatura. En esta tabla, además, se puede encontrar el peso que tendrá cada actividad evaluable en la nota final de la asignatura.

La fórmula para calcular la **nota final de la asignatura (NF)** es la siguiente:

$$NF = 0.55NT + 0.30NP + 0.15NE \text{ si } NT \geq 4, NP \geq 5 \text{ y } NE \geq 3$$

NF = 0, e.o.c.

Observaciones:

- Si un alumno obtiene una nota de teoría (NT) inferior a 4 en la convocatoria ordinaria, o una nota inferior a 5 en la nota del proyecto de programación (NP) o una nota inferior a 3 en el ejercicio del examen (NE), suspende la asignatura en la convocatoria ordinaria.
- Si un alumno suspende la asignatura en la convocatoria ordinaria, las NT y la NP aprobadas (nota ≥ 5) se guardarán para la convocatoria extraordinaria.
- **Si un alumno es repetidor del curso 2017/2018 o anteriores y tiene aprobadas las prácticas, pero desea mejorar su nota en estas actividades evaluables, lo podría hacer realizando las actividades que se propongan. Sin embargo, en el momento que entregue el primer proyecto de programación, perderá la NP obtenidas en el anterior curso académico.**

El Sistema de evaluación mediante sólo prueba final sólo se ofrecerá si así lo exige la Normativa Reguladora de los Sistemas de Evaluación en la UPM que esté vigente en el curso académico 2016-2017, y el procedimiento para optar por este sistema estará sujeto a lo que establezca en su caso Jefatura de Estudios de conformidad con lo que estipule dicha Normativa.

3. Convocatoria Extraordinaria

En el caso de que el alumno no apruebe la asignatura siguiendo uno de estos dos itinerarios, dispondrá de la convocatoria extraordinaria. En esta convocatoria, el alumno tendrá que realizar las siguientes actividades evaluables salvo que esté exento de alguna de ellas porque la haya aprobado en alguna convocatoria anterior:

- **Examen final:** será un examen que constará de dos partes: 1) Un examen (NT) tendrá un peso de 55% en la nota de la asignatura, este examen podrá constar de ejercicios tipo test y de problemas; 2) Ejercicio de recuperación práctico escrito cuya nota (NE) tendrá un peso de un 15% . Todas las notas se representarán con un valor numérico entre 0 y 10. **El alumno estará exento de realizar la primera parte del examen, si**

su nota de teoría de la asignatura (NT) es mayor o igual a 5; y estará exento de realizar la segunda parte del examen, si ya tiene aprobado el ejercicio evaluable del itinerario basado en evaluación continua o el ejercicio práctico del examen final del itinerario basado en un examen final.

- **Proyectos de programación:** serán ejercicios que se realizarán individualmente o en parejas. Se entregarán a través de un sistema de entrega web en el periodo que se establezca. A partir de las calificaciones obtenidas en estos ejercicios se obtendrá la **nota de prácticas (NP)** de la asignatura, que será un valor numérico entre 0 y 10.

La fórmula para calcular la **nota final de la asignatura (NF)** es la siguiente:

$$NF = 0.55NT + 0.15NE + 0.3NP + 0.10NPO, \text{ si } NT \geq 3, NE \geq 4 \text{ y } NP \geq 5$$

$$NF = 0, \text{ e.o.c.}$$

Observaciones:

- Si un alumno obtiene una nota en el examen final inferior a 4 en esta convocatoria, o una nota de prácticas NP inferior 5, suspende la asignatura.
- **Si un alumno suspende la asignatura en la convocatoria extraordinaria, no se guarda ninguna nota, ya que los proyectos de programación son actividades formativas que se desarrollan acorde al desarrollo de la asignatura.**
- **Si un alumno es repetidor del curso 2017/2018 o anteriores y tiene aprobadas las prácticas,** pero desea mejorar su nota en estas actividades evaluables, lo podría hacer realizando las actividades que se propongan. Sin embargo, en el momento que entregue el primer proyecto de programación perderá la NP obtenidas en el anterior curso académico.
- La práctica opcional no se puede entregar en la convocatoria extraordinaria, pero si se aprobó en la convocatoria ordinaria, se guardará su nota (NPO).

NOTA: En la tabla de actividades evaluables se supera el 100%, ya que se proporciona una actividad voluntaria que permitirá al alumno complementar su nota y poder optar a la matrícula de honor.

8. Recursos didácticos

8.1. Recursos didácticos de la asignatura

Nombre	Tipo	Observaciones
J. Chase, J. Lewis (2006) Estructura de Datos con Java. Pearson Educación	Bibliografía	
L. Craig (2004) UML y Patrones. Prentice Hall	Bibliografía	
N. Dale, H.M. Walker (1996) Abstract data types: specifications, implementations, and applications. D. C. Heath and Company	Bibliografía	
B. Eckel (2006) Thinking In Java 4th edition. Prentice Hall	Bibliografía	
M.T. Goodrich, R. Tamassia (2005) Data Structures and Algorithms in Java (4th Ed.). John Wiley and Sons	Bibliografía	
C. S. Horstmann, G. Cornell (2006) Core Java 2 (J2SE 5.0) Volumen I- Fundamentos. Prentice Hall	Bibliografía	
J. Lewis, J. Chase (2006) Estructura de datos con Java. Diseño de estructuras y algoritmos. Addison Wesley	Bibliografía	
B. Meyer (2000) Object Oriented Software Construction. Prentice Hall	Bibliografía	
G.J. Myers (2004) The Art of Software Testing, Second Edition. John Wiley & Sons	Bibliografía	

K. Sierra, B. Bates (2005) Head First Java, Second Edition. OReilly	Bibliografía	Disponible en línea desde IPs de la UPM en: http://proquest.safaribooksonline.com/book/programming/java/0596009208
Aula virtual de la UPM	Recursos web	https://moodle.upm.es/titulaciones/oficiales/course/view.php?id=6288
Salas informáticas de la escuela	Equipamiento	

9. Otra información

9.1. Otra información sobre la asignatura