



Grado en Ingeniería Informática

DATOS DE LA ASIGNATURA

Nombre:

Metodología de la Programación

Denominación en inglés:

Programming Methodology

Código:

606010201

Carácter:

Obligatorio

Horas:

	Totales	Presenciales	No presenciales
Trabajo estimado:	150	60	90

Créditos:

Grupos reducidos				
Grupos grandes	Aula estándar	Laboratorio	Prácticas de campo	Aula de informática
3	0	0	0	3

Departamentos:

Tecnologías de la Información

Áreas de Conocimiento:

Ciencia de la Computación e Inteligencia Artificial

Curso:

2º - Segundo

Cuatrimestre:

Primer cuatrimestre

DATOS DE LOS PROFESORES

Nombre:

*Márquez Hernández,
Antonio Ángel

E-Mail:

amarquez@dti.uhu.es

Teléfono:

959217388

Despacho:

TUPB-49

*Profesor coordinador de la asignatura

1. Descripción de contenidos

1.1. Breve descripción (en castellano):

- Diseño modular y creación de bibliotecas.
- Herramientas de depuración, pruebas y validación.
- Gestión de errores.
- Proyectos informáticos de programación.
- Orientación a Objetos.
- Polimorfismo, Herencia, Objetos complejos
- Persistencia de los objetos.
- Patrones para la resolución de problemas.
- Programación orientada a aspectos.

1.2. Breve descripción (en inglés):

- Modular Design
- Debug tools. Testing and validation
- Exception Management
- Object oriented Concepts
- Object Persistancy
- Introduction to Patterns
- Aspect Oriented Programing

2. Situación de la asignatura

2.1. Contexto dentro de la titulación:

Asignatura obligatoria de 2º de Grado en Ingeniería Informática adscrita al Departamento de Tecnologías de la Información en el Área de Ciencias de la Computación e Inteligencia Artificial.

Esta asignatura ubicada en el primer cuatrimestre de segundo curso requiere de los conocimientos adquiridos en las asignaturas de primer curso:

- Fundamentos de Programación
- Fundamentos de Análisis de Algoritmos
- Estructuras de Datos I

Finalmente, los conocimientos adquiridos en esta asignatura sirven de base para las asignaturas de segundo y tercer curso:

- Estructuras de Datos II
- Principios y Fundamentos de Ingeniería del Software
- Programación Concurrente y Distribuida

2.2. Recomendaciones:

Se recomienda haber cursado las asignaturas de primer curso indicadas en el apartado anterior antes de comenzar esta asignatura.

3. Objetivos (Expresados como resultados del aprendizaje):

- Desarrollar una aproximación disciplinada a la especificación, implementación, verificación y documentación de programas.
- Aprender el papel central que juega la abstracción en la tarea de programar.
- Conocer estructuras de datos, algoritmos y esquemas de uso general.
- Proporcionar los fundamentos teóricos y prácticos básicos para cursar posteriores estudios en programación.
- Introducir el paradigma de la programación orientada a objetos.
- Aprender lenguajes de programación orientados a objetos.

4. Competencias a adquirir por los estudiantes

4.1. Competencias específicas:

- **CC05:** Conocimiento, administración y mantenimiento de sistemas, servicios y aplicaciones informáticas.
- **CC06:** Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos
- **CC07:** Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema
- **CC08:** Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados

4.2. Competencias básicas, generales o transversales:

- **CB1:** Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio
- **CB5:** Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía
- **T02:** Conocimiento y perfeccionamiento en el ámbito de las TIC's

5. Actividades Formativas y Metodologías Docentes

5.1. Actividades formativas:

- Sesiones de Teoría sobre los contenidos del Programa.
- Sesiones de Resolución de Problemas.
- Sesiones Prácticas en Laboratorios Especializados o en Aulas de Informática.
- Actividades Académicamente Dirigidas por el Profesorado: seminarios, conferencias, desarrollo de trabajos, debates, tutorías colectivas, actividades de evaluación y autoevaluación.

5.2. Metodologías docentes:

- Clase Magistral Participativa.
- Desarrollo de Prácticas en Laboratorios Especializados o Aulas de Informática en grupos reducidos.
- Resolución de Problemas y Ejercicios Prácticos.
- Tutorías Individuales o Colectivas. Interacción directa profesorado-estudiantes.
- Planteamiento, Realización, Tutorización y Presentación de Trabajos.
- Conferencias y Seminarios.
- Evaluaciones y Exámenes.

5.3. Desarrollo y justificación:

Clases teóricas y de problemas de aula

Las clases de teoría se impartirán combinando clases magistrales y problemas al total de alumnos del grupo de teoría. Durante estas sesiones el alumno adquirirá los conceptos teóricos, apoyados con ejemplos y ejercicios aclaratorios, que le capacitarán para alcanzar las competencias establecidas en el apartado correspondiente de competencias de esta guía. Para el desarrollo de las clases se utilizarán transparencias apoyadas con el uso de la pizarra cuando sea necesario y ordenador portátil con retroproyector para la compilación y demostración de programas.

Clases de Laboratorio

Las clases prácticas se impartirán en el laboratorio utilizando un ordenador personal para cada alumno, del total de grupo de prácticas. En estas sesiones cada alumno de forma individual deberá implementar diferentes problemas utilizando Java y/o C++ como lenguaje de programación. El objetivo de estas sesiones es enfrentar al alumno con un desarrollo práctico que le permitirá adquirir las competencias procedimentales específicas, así como la mayoría de las competencias transversales.

Clases de problemas

Las sesiones de problemas pueden tener dos formatos:

1. **En grupo:** se realizarán de forma que se dividan en varios grupos que trabajarán sobre un mismo problema. En estas sesiones se planteará un problema de programación y cada grupo debe dar una solución al mismo.
2. **Individuales:** Se plantearán problemas cuya solución el alumno puede entregar por escrito para que sea tenido en cuenta en su evaluación. Estos ejercicios puntuables no tienen consideración de exámenes pudiendo ser realizados sin preaviso y siempre en horario de clase. Su principal finalidad es complementar las evaluaciones de aquellos alumnos de los que no se disponga evaluaciones por participación en problemas de grupo.

Exposiciones y Seminarios

Para que los alumnos conozcan aspectos avanzados de lenguajes de programación OOP se podrán impartir seminarios sobre las alternativas existentes.

6. Temario desarrollado:

Tema 0: Introducción Conceptos Avanzados de Orientación a Objetos

- 0.1 Herencia.
- 0.2 Visibilidad.
- 0.3 Polimorfismo y otros conceptos.

Tema 1: Herencia y Polimorfismo en C++

- 1.1 Herencia simple vs múltiple. Redefinición de métodos.
- 1.2 Jerarquías de clases y visibilidad.
- 1.3 Polimorfismo y ligadura dinámica: métodos virtuales.
- 1.4 Tipos de constructores. Destruyores polimórficos. Funciones amigas.
- 1.5 Excepciones OOP.
- 1.6 Manejo de ficheros OOP.
- 1.7 Genéricos.

Tema 2: Herencia y Polimorfismo en Java

- 2.1 Herencia.
- 2.2 Jerarquías de clases y visibilidad.
- 2.3 Polimorfismo.
- 2.4 Tipos de constructores y destructores polimórficos.
- 2.5 Excepciones OOP.
- 2.6 Manejo de ficheros OOP.
- 2.7 Genéricos.

Tema 3: Patrones simples en Java

- 3.1 Diseño de tipos.
- 3.2 Contratos, implementación y reutilización.
- 3.3 Factorías e iterables. Colecciones.

Tema 4: Librerías y herramientas en Java

- 4.1 Librerías gráficas.
- 4.2 Generación de documentación.
- 4.3 Generación de archivos Jar.

Seminarios:

Otras Técnicas/Lenguajes/Plataformas orientados a objetos.

7. Bibliografía

7.1. Bibliografía básica:

C++

Stroustrup, Bjarne. El Lenguaje de Programación C++. Ed: Addison Wesley [2002]
Deitel, Harvey M. -; Deitel, Paul J. Cómo Programar en C++ (9ª ed.). Ed: Pearson Educación, [2014]
Cohoon, James P. - Davidson, Jack W. Programación y Diseño en C++. Ed: McGraw-Hill [2000]

Java

Eckel, Bruce. Piensa en Java. Ed: Pearson-Prentice Hall, [2007]
Deitel, Paul J - Deitel, Harvey M. Cómo Programar en Java. Ed: Pearson Educación, [2008]
Arnold, Ken - Gosling, J. - Colmes, D. El Lenguaje de Programación Java. Ed: Pearson Educación, [2001]

7.2. Bibliografía complementaria:

Weiss, Mark Allen. Estructuras de Datos en Java. Ed: Addison Wesley, [2007]
Aho, Alfred V. - Hopcroft, J. - Ullman, J. D. Estructuras de Datos y Algoritmos. Ed: Addison Wesley. [1988]

8. Sistemas y criterios de evaluación.

8.1. Sistemas de evaluación:

- Examen de teoría/problemas
- Defensa de Prácticas
- Defensa de Trabajos e Informes Escritos
- Examen de prácticas

8.2. Criterios de evaluación y calificación:

Criterios de Evaluación

Febrero:

Aspecto.....	Criterio.....	Instrumento.....	Peso.....
Conocimientos Teóricos.....	<i>Exposición escrita de conocimientos.....</i>	Examen de Teoría Escrito.....	50%
	<i>y su aplicación a distintas situaciones</i>		
Conocimientos Prácticos.....	<i>Capacidad de implementar programas.....</i>	Examen de Prácticas reducido en C++ y Java.....	30%
	<i>en C++ y Java</i>		
Conocimientos Prácticos.....	<i>Calidad de los trabajos prácticos.....</i>	Entrega y Defensa de Prácticas.....	10%
Participación y Adquisición... de Competencias.....	<i>Participación activa en las sesiones.....</i> <i>aportando soluciones válidas a los</i> <i>problemas y asimilando los criterios</i>	Defensa de Trabajos (Sesiones de Problemas)...	10%

Septiembre:

Aspecto.....	Criterio.....	Instrumento.....	Peso.....
Conocimientos Teóricos.....	<i>Exposición escrita de conocimientos.....</i>	Examen de Teoría Escrito.....	60%
	<i>y su aplicación a distintas situaciones</i>		
Conocimientos Prácticos.....	<i>Capacidad de implementar programas.....</i>	Examen de Prácticas completo en C++ o Java....	40%
	<i>en C++ y Java</i>		

Evaluación parte Teórica

Convocatoria de Febrero (Peso 50%):

- El examen Teórico consistirá en una prueba escrita compuesta por un test, problemas prácticos y opcionalmente preguntas cortas de teoría.
- En el caso de existir test cada pregunta acertada se puntuará positivamente y cada pregunta errónea negativamente de forma proporcional al número de respuestas, no puntuando las no contestadas.
- El examen podrá incluir preguntas y/o problemas sobre los seminarios, actividades de laboratorio y actividades propuestas por el profesor durante el curso.

Convocatoria de Septiembre (Peso 60%):

- Existirá un examen final de teoría para aquellos alumnos que tengan suspensa la parte Teórica:
(Examen Teórico Escrito x 50% + Sesiones de Problemas x 10%) / 60% < 5

Evaluación parte Práctica

Convocatoria de Febrero (Peso 30%):

- Las prácticas (2 Actividades Académicamente Dirigidas) se entregan en fecha y forma, realizándose un examen de modificación reducido sobre cada una de ellas.
- Aquellos alumnos que **no entreguen las prácticas no podrán realizar el examen de Prácticas reducido**.
- Tener una asistencia mínima del 80% a prácticas y presentarse a los 2 exámenes de modificación (examen de Prácticas reducido).

Convocatoria de Septiembre (Peso 40%):

- Existirá un examen final de prácticas (examen de Prácticas completo) para aquellos alumnos que tengan el examen de Prácticas reducido suspenso (calificación < 5 sobre 10)

Evaluación de Entrega y Defensa de Prácticas

Convocatoria de Febrero (Peso 10%):

- El profesor evaluará la calidad de los trabajos prácticos, considerando la solución de código propuesta y la claridad y eficiencia de la implementación desarrollada.

Convocatoria de Septiembre (Peso 0%):

- El valor de estas actividades NO se guardará para esta convocatoria ya que son actividades realizadas durante el periodo de docencia donde el alumnado tiene que entregar las prácticas propuestas.

Evaluación de Trabajos (Sesiones de Problemas)

Convocatoria de Febrero (Peso 10%):

- El profesor evaluará la participación de cada alumno en clase, considerando la solución propuesta a los problemas resueltos en clase (en grupos o individualmente) y la capacidad de crítica y resolución a los posibles fallos en las soluciones

planteadas.

Convocatoria de Septiembre (Peso 0%):

- El valor de estas actividades NO se guardará para esta convocatoria ya que son actividades realizadas durante el periodo de docencia donde el alumnado tiene que participar activamente.

La nota del alumno será:

Si (Examen Teórico Escrito x 50% + Sesiones de Problemas x 10%) / 60% >= 5

Nota Acta = Examen Teoría x 50% + Sesiones Problemas x 10% + Examen Prácticas x 30% + Entrega y Defensa Prácticas x 10%

En caso contrario, **si (Examen Teórico Escrito x 50% + Sesiones de Problemas x 10%) / 60% < 5**

Nota Acta = (Examen de Teoría x 50% + Sesiones de Problemas x 10%) / 60%

Para aplicar la fórmula anterior es necesario tener aprobado el examen de Prácticas (calificación >=5 sobre 10)

En caso contrario el alumno tendrá SUSPENSA la asignatura, independientemente de la nota que tenga en el Examen de Teoría + Sesiones de Problemas.

Criterio para guardar las notas para la convocatoria de septiembre:

Sesiones de Problemas y Defensa de Prácticas:

- La nota de las Sesiones de Problemas y Defensa de Prácticas NO se guardan para la convocatoria de septiembre, ya que son actividades realizadas durante el periodo de docencia donde el alumnado tiene que participar activamente.

Examen de Teoría:

- Para ver si se conserva la nota del examen de Teoría hay que tener en cuenta tanto la nota sacada en dicho examen como la nota de las Sesiones de Problemas (una y otra se compensan según su peso, de forma que hay que tener en cuenta ambas).
- De esta forma, la nota del examen de Teoría se guarda hasta la convocatoria de septiembre inclusive, siempre y cuando el alumno tenga una calificación >= 5 (sobre 10) en la siguiente fórmula:

Si (Examen Teórico Escrito x 50% + Sesiones de Problemas x 10%) / 60% >= 5

La nota resultante de dicha fórmula será la considerada obtenida en septiembre (pondera un 60%)

En caso contrario

La nota de Teoría no se conserva para Septiembre (parte teórica suspensa)

Examen de Prácticas:

- La nota del examen de Prácticas se guarda hasta la convocatoria de septiembre inclusive, siempre y cuando el alumno tenga una calificación >= 5 (sobre 10), no guardándose para convocatorias posteriores a la de septiembre.
- En ese caso, la nota considerada obtenida en septiembre vendrá dada por la siguiente fórmula:
(Examen de Prácticas x 30% + Entrega y Defensa de Prácticas x 10%) / 40% (pondera un 40%)

En cualquier caso, ninguna nota (teórica, práctica, de problemas y defensa de prácticas) de un curso académico se guarda para cursos académicos posteriores.

9. Organización docente semanal orientativa:

	Semanas	Grupos Grandes	Grupos Reducidos Aula Estándar	Grupos Reducidos Aula de Informática	Grupos Reducidos Laboratorio	Grupos Reducidos prácticas de campo	Pruebas y/o actividades evaluables	Contenido desarrollado
#1	2	0	2	0	0		Presentación. Tema 0	
#2	2	0	2	0	0		Tema 1	
#3	2	0	2	0	0		Tema 1	
#4	2	0	2	0	0		Tema 1	
#5	2	0	2	0	0		Tema 1	
#6	2	0	2	0	0	Sesión de Problemas	Tema 1	
#7	2	0	2	0	0		Tema 1	
#8	2	0	2	0	0		Tema 2	
#9	2	0	2	0	0	Entrega AAD C++	Tema 2	
#10	2	0	2	0	0	Prueba Laboratorio C++	Tema 2	
#11	2	0	2	0	0		Tema 2	
#12	2	0	2	0	0	Sesión de Problemas	Tema 3	
#13	2	0	2	0	0		Tema 3	
#14	2	0	2	0	0	Entrega AAD Java	Tema 3	
#15	2	0	2	0	0	Prueba Laboratorio Java	Tema 4	
	30	0	30	0	0			