

**Guía docente de la asignatura**

Asignatura	FUNDAMENTOS DE PROGRAMACIÓN		
Materia	FUNDAMENTOS BÁSICOS DE INFORMÁTICA		
Módulo	FUNDAMENTOS BÁSICOS		
Titulación	GRADO EN INGENIERÍA INFORMÁTICA / DOBLE GRADO INFORMÁTICA+ESTADÍSTICA (INDat)		
Plan	545 / 551	Código	46904
Periodo de impartición	1er. CUATRIMESTRE	Tipo/Carácter	FB (Formación Básica)
Nivel/Ciclo	GRADO	Curso	1º
Créditos ECTS	6 ECTS		
Lengua en que se imparte	CASTELLANO		
Profesor/es responsable/s	María Luisa González Díaz Esperanza Manso Martínez Alma María Pisabarro		
Datos de contacto (E-mail, teléfono...)	mluisa@infor.uva.es , 983 423000 ext. 5615, despacho 1D015 manso@infor.uva.es , 983 423000 ext. 5622, despacho 1D022 alma@infor.uva.es , 983 423000 ext. 5620, despacho 1D020		
Horario de tutorías	Véase www.uva.es → Alumnos → Apoyo → Tutorías		
Departamento	INFORMÁTICA (ATC, CCIA, LSI)		

1. Situación / Sentido de la Asignatura

1.1 Contextualización

Contexto profesional

El título está orientado a la consecución de capacidades que habiliten para el desempeño de calidad, con carácter general, de la profesión de ingeniería informática en todos sus ámbitos de actuación, y para tareas de desarrollo de software, sistemas y servicios en el ámbito de la ingeniería informática. Aún cuando no es posible saber cuáles serán los métodos, técnicas, lenguajes etc. que se vayan a usar en un futuro, el dominio de los conceptos fundamentales de programación serán imprescindibles en cualquier tarea relacionada con el desarrollo y mantenimiento del software.

Contexto curricular:

Pertenece a la materia FUNDAMENTOS DE INFORMÁTICA en el bloque formativo FUNDAMENTOS BÁSICOS, que lleva asociadas todas las competencias de formación básica. La materia se desarrolla en los dos primeros semestres, siendo esta asignatura junto con Sistemas Digitales las únicas de la materia desarrolladas en el primer semestre. Se trata por lo tanto del primer encuentro del alumno con la Informática (en una titulación del mismo nombre), en su aspecto "software". Las competencias que tiene asignadas en cuanto a su carácter de formación básica se refieren tanto a la capacidad de comprender y dominar los conceptos básicos que son relevantes en cualquier desempeño de la profesión, como naturalmente al uso de software como herramienta de solución de problemas.

Se imparte simultáneamente con las asignaturas como "Fundamentos de Organización de Empresas", "Fundamentos de Matemáticas", "Matemática Discreta" y "Sistemas Digitales" (todas ellas de Formación Básica). En especial compartirá con la asignatura "Matemática Discreta" algunos elementos generales (lógica, relaciones, grafos, ...), que serán estudiados en profundidad allí y aplicados aquí. La asignatura de "Sistemas Digitales" proporcionará una visión útil para la comprensión del concepto abstracto de máquina que se usa en esta asignatura (circuitos, diseño modular).

1.2 Relación con otras materias

En la temática, guarda una fuerte relación con otras asignaturas de la titulación:

Especialmente con las asignaturas relacionadas con la programación propiamente dicha y con las estructuras de datos y algoritmos, a las que sirve de inicio, y que completarán los conceptos. En particular es precursora de "Paradigmas de Programación" puesto que "Fundamentos de Programación" proporcionará el primer lenguaje y con ello el primer paradigma.

Asignaturas de mención que se refieren a lenguajes de programación y en general lenguajes formales.

Las relacionadas con Análisis y Diseño de Software, e Ingeniería de Software, o Interacción Persona-Computadora, en cuanto a que es el más rudimentario y primitivo ejemplo de estos procesos. Por otra parte, estas asignaturas enmarcarán el proceso de la programación en su lugar adecuado dentro de otro mucho más general.

Las de Computadoras, Redes y Sistemas Operativos, que no sólo requerirán cierto conocimiento de programación en general, sino que necesitan que el alumno sepa utilizar un lenguaje de programación concreto con alguna soltura.

En general, los conceptos que proporciona la programación están implícitos en muchas disciplinas informáticas. Se han reseñado las más importantes.

1.3 Prerrequisitos

En realidad, no se necesita más requisito formativo que una cierta estructuración lógica del pensamiento, ni más actitud que el interés por la materia, para alcanzar la destreza que se pretende en la asignatura, por medio de una cantidad de trabajo razonable.

2. Competencias

2.1 Generales

Código	Descripción
CG1	Capacidad para concebir, redactar, organizar, planificar, desarrollar y firmar proyectos en el ámbito de la ingeniería en informática que tengan por objeto, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación especificadas en la sección 3.1 de la memoria de Grado, la concepción, el desarrollo o la explotación de sistemas, servicios y aplicaciones informáticas..
CG3	Capacidad para diseñar, desarrollar, evaluar y asegurar la accesibilidad, ergonomía, usabilidad y seguridad de los sistemas, servicios y aplicaciones informáticas, así como de la información que gestionan
CG4	Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación específicas indicadas en la sección 3.1 de la memoria de Grado
CG5	Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería de software como instrumento para el aseguramiento de su calidad, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de formación específicas indicadas en la sección 3.1 de la memoria de Grado
CG10	Conocimientos para la realización de mediciones, cálculos, valoraciones, tasaciones, peritaciones, estudios, informes, planificación de tareas y otros trabajos análogos de informática, de acuerdo con los conocimientos adquiridos según lo establecido en las competencias de específicas indicadas en la sección 3.1 de la memoria de Grado

2.2 Transversales

Código	Descripción
CT1	Capacidad de análisis y síntesis
CT2	Capacidad de organizar y planificar
CT3	Comunicación oral y escrita en la lengua propia
CT4	Capacidad para la lectura de textos técnicos en inglés
CT5	Habilidades de gestión de la información
CT6	Resolución de problemas
CT7	Toma de decisiones
CT8	Capacidad crítica y autocrítica
CT9	Trabajo en equipo
CT10	Capacidad de trabajar en un equipo interdisciplinar
CT11	Responsabilidad y compromiso ético
CT12	Liderazgo
CT13	Capacidad de aplicar los conocimientos en la práctica
CT14	Capacidad de aprender
CT15	Capacidad de adaptarse a nuevas situaciones
CT16	Habilidad para trabajar de forma autónoma
CT17	Iniciativa y espíritu emprendedor

2.3 Específicas de formación básica



Código	Descripción
FB3	Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para la resolución de problemas propios de la ingeniería
FB4	Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería
FB7	Capacidad de transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado

2.4 Específicas Comunes a la rama de Informática

Código	Descripción
CI7	Conocimiento, diseño y utilización de forma eficiente de los tipos y estructuras de datos más adecuados a la resolución de un problema.

3. Objetivos

O1Met	Conocer y saber aplicar las bases metodológicas de la programación a la construcción de programas de calidad, que sean sencillos, eficaces y fáciles de entender y probar
O2Flj	Conocer, caracterizar y saber emplear las estructuras básicas de programación en la construcción modular de programas
O3Tip	Conocer la estructura, características y casos de aplicación de los tipos de datos estructurados más importantes
O4Din	Saber emplear de forma adecuada y eficiente estructuras de datos dinámicas en la construcción de programas
O5Prb	Abordar la prueba y depuración de programas siguiendo las estrategias más adecuadas

**4. Tabla de dedicación del estudiante a la asignatura**

ACTIVIDADES PRESENCIALES	HORAS	ACTIVIDADES NO PRESENCIALES	HORAS
Clases teórico-prácticas (T/M)	28	Estudio y trabajo autónomo individual	70
Clases prácticas de aula (A)			
Laboratorios (L)	22	Estudio y trabajo autónomo grupal	20
Prácticas externas, clínicas o de campo			
Seminarios (S)	4		
Tutorías grupales (TG)			
Evaluación (fuera del periodo oficial de exámenes)	6		
Total presencial	60	Total no presencial	90





5. Bloques temáticos

Bloque 1: Conceptos básicos

Carga de trabajo en créditos ECTS:

a. Contextualización y justificación

Es imprescindible comenzar abordando los conceptos más básicos de programación. Se explican aquí marcando las ideas de “datos” y “algoritmos”.

b. Objetivos de aprendizaje

Cubrir en especial el objetivo O1Met y los rudimentos de O3Tip

c. Contenidos

Tema 1.1: Entorno de la programación
Tema 1.2: Variables. Tipos básicos
Tema 1.3: Métodos

d. Métodos docentes

Ver anexo (8).

e. Plan de trabajo

En las clases teóricas se desarrollarán los conceptos y se realizarán ejemplos ilustrativos. En las clases de laboratorio asociadas, los alumnos implementarán programas especificados para ello. Como trabajo personal, los alumnos deberían resolver ejercicios y realizar por su cuenta programas que requieran solamente los conceptos estudiados, de la lista de ejercicios a las que tendrán acceso en la página de la asignatura y en los libros referidos.

f. Evaluación

Ver punto 7 de esta guía.

g. Bibliografía básica

- Capítulos 1, 2 y 3 de “How to Think as a Computer Scientist (Java version)” (Downey)
- Capítulos 1, 2 y 4 de “Introduction to Programming using Java” (Eck)

Ambos descargables vía web

h. Bibliografía complementaria

i. Recursos necesarios

Página de la asignatura en entorno Moodle (www.uva.es/campusvirtual o aulas.inf.uva.es)

Software: Editor de textos (vim). Compilador de Java y máquina virtual Java (javac, java)

Bloque 2: Estructuras de controlCarga de trabajo en créditos ECTS: **a. Contextualización y justificación**

Tratando de reforzar la idea de programa en cuanto a sus facetas “algoritmo + estructura de datos”, este bloque se centra en los rudimentos de los algoritmos. Evidentemente se utilizarán y completarán los tipos de datos básicos.

b. Objetivos de aprendizaje

Cubrir en especial el objetivo O2Flj y parte de O5Prb

c. Contenidos

Tema II.1: Estructuras alternativas
Tema II.2: Recursión
Tema II.3: Iteración

d. Métodos docentes

Ver anexo (8)

e. Plan de trabajo

En las clases teóricas se desarrollarán los conceptos y se realizarán ejemplos ilustrativos. En las clases de laboratorio asociadas, los alumnos implementarán programas especificados para ello. Como trabajo personal, los alumnos deberían resolver ejercicios y realizar por su cuenta programas que requieran solamente los conceptos estudiados, de la lista de ejercicios a las que tendrán acceso en la página de la asignatura y en los libros referidos.

f. Evaluación

Ver punto 7 de esta guía

g. Bibliografía básica

- Capítulos 4 a 7 de “How to Think as a Computer Scientist (Java version)” (Downey)
- Capítulos 3 y 4 de “Introduction to Programming using Java” (Eck)

Ambos descargables vía web

h. Bibliografía complementaria**i. Recursos necesarios**

Página de la asignatura en entorno Moodle (www.uva.es/campusvirtual o aulas.inf.uva.es)

Software: Editor de textos (vim). Compilador de Java y máquina virtual Java (javac, java). Entorno de desarrollo (Eclipse)

Bloque 3: Tipos de datos estructuradosCarga de trabajo en créditos ECTS: **a. Contextualización y justificación**

Bloque dedicado a datos estructurados y a introducir los algoritmos básicos relacionados.

b. Objetivos de aprendizaje

Completar el objetivo O3Tip y cubrir O4Din

c. Contenidos

Tema III.1 Tipos multidimensionales
Tema III.2 Objetos
Tema III.3 Tipos dinámicos elementales
Tema III.4 Ficheros

d. Métodos docentes

Ver anexo (8)

e. Plan de trabajo

En las clases teóricas se desarrollarán los conceptos y se realizarán ejemplos ilustrativos. En las clases de laboratorio asociadas, los alumnos implementarán programas especificados para ello. Como trabajo personal, los alumnos deberían resolver ejercicios y realizar por su cuenta programas que requieran solamente los conceptos estudiados, de la lista de ejercicios a la que tendrán acceso en la página de la asignatura y en los libros referidos.

f. Evaluación

Ver punto 7 de esta guía

g. Bibliografía básica

- Capítulos 8 a 14 de "How to Think as a Computer Scientist (Java version)" (Downey)
- Capítulos 5, 7, 9, 11 de "Introduction to Programming using Java" (Eck)

Ambos descargables vía web

h. Bibliografía complementaria**i. Recursos necesarios**

Página de la asignatura en entorno Moodle (www.uva.es/campusvirtual o aulas.inf.uva.es)

Software: Entorno de desarrollo (Eclipse)

**Bloque 4: (L) Herramientas de programación (Laboratorio)**

Carga de trabajo en créditos ECTS: 0'8

a. Contextualización y justificación

Las prácticas requieren el conocimiento de un entorno real de programación, incluyendo una introducción al uso de un Sistema Operativo y a mecanismos de desarrollo y documentación. En este bloque se imparten los contenidos cuyo mejor alojamiento se encuentra en el entorno del laboratorio y se desarrollan de forma práctica los contenidos impartidos en el aula.

b. Objetivos de aprendizaje

Práctica real de los bloques 1, 2 y 3

c. Contenidos

- Sistema operativo. Edición-compilación-ejecución
- Entrada y salida en Java
- Entorno de Desarrollo Integrado (IDE)
- Acceso a documentación de Java. Bibliotecas de clases

d. Métodos docentes

Ver anexo (8)

e. Plan de trabajo

Explicación y uso de conceptos y herramientas de forma guiada en laboratorio durante las sesiones semanales de laboratorio, de 2 horas cada una. Las herramientas se continuarán utilizando y afianzando en las sesiones de laboratorio correspondientes a los bloques 1, 2 y 3.

f. Evaluación**g. Bibliografía básica**

- Guiones de laboratorio (página de la asignatura)
- Secciones 2.4 y 2.6 de "Introduction to Programming using Java" (Eck)
- Apéndices B y C de "How to Think as a Computer Scientist" (Downey)

h. Bibliografía complementaria

- Página web download.oracle.com
- Página web www.eclipse.org

i. Recursos necesarios

Acceso al sistema operativo Linux.

Página de la asignatura en entorno Moodle (www.uva.es/campusvirtual o aulas.inf.uva.es)

Software: Editor (vim). Compilador de Java y máquina virtual Java (javac, java). Entorno de desarrollo (Eclipse)

**Bloque 5: (S) Calidad (Seminarios)**Carga de trabajo en créditos ECTS: **a. Contextualización y justificación**

Bloque diseñado para que el alumno adquiriera una visión global de la programación, comprobando en la práctica el significado e importancia de todo el ciclo de vida del software (a pequeña escala) y la necesidad de atender a criterios de calidad.

b. Objetivos de aprendizaje

Completar los objetivos O1Met y O5Prb

c. Contenidos

S1. Pruebas y depuración
S2. Criterios de calidad externos

d. Métodos docentes

Ver anexo (8)

e. Plan de trabajo

Se desarrollarán 2 sesiones de prácticas específicas en el laboratorio, con evaluación, autoevaluación y evaluación cruzada de programas.

f. Evaluación

Ver punto 7 de esta guía

g. Bibliografía básica

- Guiones de laboratorio (en la página de la asignatura)

h. Bibliografía complementaria

- "How to Think as a Computer Scientist (Java version)" (Downey)
- "Introduction to Programming using Java" (Eck)

i. Recursos necesarios

Acceso a sistema operativo Linux.

Página de la asignatura en entorno Moodle (www.uva.es/campusvirtual o aulas.inf.uva.es)

Acceso a Evalcomix en el entorno Moodle

Software: Editor (vim). Compilador de Java y máquina virtual Java (javac, java). Entorno de desarrollo (Eclipse)

6. Temporalización (por bloques temáticos)

BLOQUE TEMÁTICO	CARGA ECTS	PERIODO PREVISTO DE DESARROLLO
Bloque 1: I. Conceptos básicos	0'7 ECTS	Semanas 1 a 3 aprox.
Bloque 2: II. Estructuras de control	1'5 ECTS	Semanas 4 a 8 aprox.
Bloque 3: III. Tipos de datos estructurados	2'2 ECTS	Semanas 9 a 15 aprox.
Bloque 4: (L) Herramientas de programación	0'8 ECTS	Todo el semestre, laboratorios
Bloque 5: (S) Calidad	0'8 ECTS	Semanas 7 y 12 aprox.

7. Sistema de calificaciones – Tabla resumen

INSTRUMENTO/PROCEDIMIENTO	PESO EN LA NOTA FINAL	OBSERVACIONES
1 evaluación en laboratorio, correspondiente al bloque 5 (ES)	10%	Sesión de laboratorio. Incluye autoevaluación y evaluación "inter pares".
2 evaluaciones en aula (EA1, EA2), individuales.	15%	Sesiones de aula. Incluirán la materia estudiada hasta ese momento.
Entrega de práctica (EP), en grupos pequeños (2-3 alumnos).	20%	Última semana del curso. Entrega en fecha común, defensa en sesiones específicas por grupos de laboratorio.
Examen final escrito (EX1), individual.	55%	Periodo de exámenes.

CRITERIOS DE CALIFICACIÓN

- **Convocatoria ordinaria:**
 - La práctica se realizará en grupos de 2 alumnos (excepcionalmente de 3). Se defenderá oralmente en sesión de entrega específica para cada grupo de laboratorio. Se valorará respecto a los criterios de calidad estudiados y el propio acto de defensa.
 - Se exige un mínimo del 36% (2 de 5'5 puntos) en la calificación del examen escrito (EX1). En caso contrario la calificación en la convocatoria será "Suspenso", con la nota numérica correspondiente a este examen.
 - Superado este mínimo, se calculará la nota del alumno según los porcentajes especificados en la tabla anterior (10% ES, 15% EA, 20% EP, 55% EX1).
- **Convocatoria extraordinaria:**
 - Se conservarán las calificaciones denominadas ES y EP obtenidas durante el curso.
 - Se realizará un nuevo examen escrito EX2, con un peso del 70% de la nota de la asignatura
 - Se exigirá un mínimo de un 40% en el examen escrito (2'8 de 7 puntos). En caso contrario la calificación en la convocatoria será "Suspenso", con la nota numérica correspondiente a este examen.
 - Superado este mínimo, la nota se calculará asignando un peso del 70% al examen escrito (EX2), un 20% a la práctica (EP), y el 10% a la calificación ES

8. Anexo: Métodos docentes

Actividad	Metodología
Clase de teoría	<ul style="list-style-type: none">• Clase magistral participativa• Estudio de ejemplos en aula• Resolución de problemas



Clase práctica	<ul style="list-style-type: none">• Clase magistral participativa• Realización de ejercicios y programas planteados por el profesorado• Autoevaluación y evaluación "inter pares"
Trabajo personal	<ul style="list-style-type: none">• Lectura de bibliografía y estudio de la materia• Lectura de programas• Realización de ejercicios y programas elegidos por el alumno, con sus pruebas y mantenimiento• Realización de un programa ("práctica") en grupos de 2-3 alumnos.

9. Anexo: Cronograma de actividades previstas

Aparecerá en el entorno Moodle de la asignatura, en aulas.inf.uva.es

